Commodore INFO

PRIJS ƒ 7,95 / Bfr. 160

8 710966 001332

ONAFHANKELIJK BLAD VOOR AMIGA- EN C64- GEBRUIKERS Jaargang 7, NO.6, sept./okt. '90



Virtual Reality

VectorTrace 1.1

Bouw een parallelle interface

Printer-problemen met GEOS

Final Chesscard voor de C-64

Interview met François Lionet (AMOS)

COLOFON

Commodore INFO is een uitgave van:
Sala Communications
Postbus 43048
1009 ZA Amsterdam

Uitgever:
V.Sharfman

Redactie:
ir. L. Sala (hoofdredacteur),
W.A. Scheer (eindredacteur),
drs. U. Schuurmans,
R.Goudriaan, B. Venema,
P. Boncz, MGCC/Johan&Johan,
Michel de Boer, Hylke Sprangers,
Rosa van Tijn
telefoon: 020-228871

Produktie:
drs. H. Zoete,
J. Broekhuizen

Advertentie-exploitatie:
ing. V. Sala, D. van Vlijmen
ing. B. Sala, H. Bia
telefoon: 020-273198

Abonnementen en administratie:
Marjo Jansen
telefoon: 020-248006

Vragen betreffende abonnementen ontvangen wij bij voorkeur schriftelijk, met meesturen van het omslagetiket.
Telefonisch uitsluitend tussen 12.00 en 15.00 uur.

Vragen over artikelen:
Voor vragen en opmerkingen over artikelen of anderszins graag een briefje aan de redactie (zie boven).

Listingtelefoon:
02155-25162
(ma: 17.00-21.00 u)

Illustraties:
Ben van Mierlo

Cover:
Commodore/Mandarin
(Amiga 500/AMOS)

Zetwerk & druk:
NDB, Zoeterwoude

Distributie:
In Nederland: Betapress, Gilze
In België: AMP, Brussel

ISSN: 0169-3085

Inhoud

Vaste rubrieken:

Redactioneel

'Voor het najaar staan bij Commodore weer heel wat op stapel. Traditioneel zien de de naam Commodore in het voetbalseizoen wat vaker, omdat men heel wat aan sportsponsoring doet. Maar ook qua marketing zien we nieuwe initiatieven, al gaat het op dit moment vooral om het positioneren van de Commodore PC lijn. Men hanteert duidelijk het prijswapen en gebruikt afzetkanalen als de Macro om grote aantallen weg te zetten. De organieke dealers van CBM zijn daar niet zo erg blij mee en we vinden dat men daarmee uit de hoek van de A-merken wegloopt. En dat hoeft toch niet, de Commodore PC's en we hebben er ondertussen heel wat getest, doen kwalitatief niet onder voor de A- merken.

De oude trouwe C-64 schijnt ook van Commodore zelf steeds minder aandacht te krijgen en voor de Amiga mikt men ook steeds hoger. Net is er een werkende 68030 Amiga, of er wordt al weer gesproken over de 68040 als de nieuwe motor, waarschijnlijk van wat dan de Amiga 4000 gaat heten. Maar logica heeft weinig met de naamgeving van de computers te maken, dus misschien wordt het gewoon de Amiga 3500. Met steeds meer power blijft de Amiga de beste machine om aan desktop video te doen. Helaas zien we qua horizontale toepassingen, dus gewoon boekhouden, tekstverwerking en database niet zo erg veel vernieuwing. De Amiga schuift op deze manier naar de hoek van de creatieve specialisten en de spelfanaten, dat is misschien jammer. Aan de andere kant, er is alles voor nog krachtiger hardware, wanneer je real-time video wilt doen. De krachtigste Amiga's worden nu gebruikt voor interessante en revolutionaire toepassingen als 'Virtual Reality', en ook de combinatie met CD-ROM in de CDTV-Amiga belooft veel. Commodore Info blijft u op de hoogte houden, al voelen we ons steeds meer balanceren tussen de wensen van onze C-64 en Amiga lezers.

Luc Sala

Excuses...

In verband met ziekte en vakantie is de aflevering Graphics op de 64, die voor dit nummer gepland was, naar nummer 7 verschoven.
De serie Beeldmanipulaties hopen we in dat nummer ook weer op te pakken met een extra lange aflevering.

De redactie

Abonnement:

Tot het einde van het jaar (2 nummers): *f* 15,- of Brf. 255
Betaling op giro 4985259 (België: BBL nr. 310050602562) t.n.v. SAC/Commodore-Info.
Oude nummers kunt u alleen krijgen bij vooruitbetaling van *f* 6,75 op de bovenstaande rekening.
Ook telefonisch bestellen is mogelijk. Bel GRATIS 06-02242222 (teleservice) elke dag van 20.20 uur (dus ook in het weekend). België: 115555, dagelijks tot 22.00 uur. Deze nummers zijn allen bedoeld voor de opgave van nieuwe abonnementen.
Opzegging dient schriftelijk te geschieden uiterlijk twee maanden voor de aanvang van een nieuwe abonnementsperiode van een jaar.

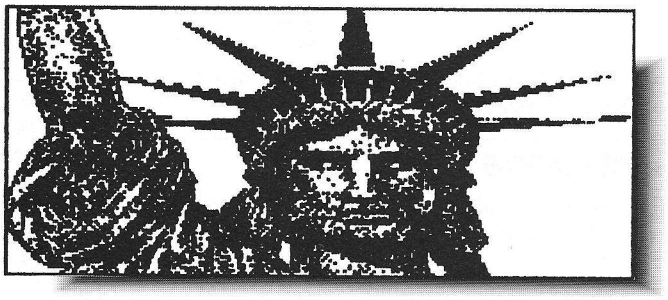
GEOS INFO

Na een reis van zo'n 7 maanden door het Caraïbisch gebied ben ik weer achter mijn PC gekropen om u op de hoogte te brengen van wat GEOS noviteiten en tips. Ik heb regelmatig de Commodore Info aan boord van ons schip mogen ontvangen en ik heb daarin gelezen dat Peter Boncz mijn taak als redacteur van de GEOS Info rubriek op een zeer goede wijze heeft waargenomen. Hiervoor wil ik hem nog hartelijk bedanken.

Ook heeft de Stichting GEOS gebruikers uit Almere behoorlijke bijdragen geleverd ten behoeve van de GEOS gebruikers, u dus! Tijdens mijn omzwervingen ben ik hier en daar wat nieuwe dingen met betrekking tot GEOS tegengekomen, die ik u niet wil onthouden.

Nieuwe GEOS diskette van Run

Terwijl ik in Fort Lauderdale (Florida) langs verschillende shops wandelde, viel mijn oog op een artikel over de nieuwste GEOS- diskette, die weliswaar niet door Berkeley Softworks wordt uitgegeven, maar door Run Magazine. Dit blad is zeer actief op het gebied van GEOS, hetgeen resulteert in het regelmatig uitbrengen van diskettes met allerlei nieuwe applicaties en utilities werkend onder het GEOS besturingssysteem. Recentelijk brachten zij weer een nieuwe diskette uit, de derde in de reeks en wel 'The GEOS Companion'-diskette. Deze collectie van creatieve GEOS gereedschappen bevat programma's voor muziek, animatie, spelletjes, clip-art en utilities. Men beweert zelfs dat deze diskette één van de beste GEOS accessoires is die ooit is uitgebracht. Inmiddels heb ik de diskette besteld en aan een nader onderzoek onderworpen.



Figuur 1

G.O Bach

De diskette, die ook nu weer als dubbelzijdige 5 disk wordt geleverd, bevat hoofdzakelijk programma's die de creativiteit van de gebruiker inspireren. Zo treft men als eerste een music-composer onder de naam G.O. Bach aan, een electronische music synthesizer, waarmee eigen composities en interessante sound-effecten kunnen worden gegenereerd. Nu zult u zeggen: 'Er bestaan toch zeker al genoeg music-composer programma's?' Maar G.O. Bach onderscheidt zich toch enigzins van de reguliere muziekprogramma's voor de Commodore 64.
Het programma bestaat uit 3 ontwerp-schermen, allemaal met de gebruikelijke GEOS scherm lay-out. Het programma wordt, zoals alle programma's op deze diskette, geleverd in een 40-koloms en 80 koloms (C128)-versie. De schermen hebben allen een professioneel uiterlijk, met schuifregelaars en schakelaars die omgezet kunnen worden. Het werken met de G.O. Bach vereist wel enige kennis van muziek componeren, maar de handleiding legt in ieder geval duidelijk uit hoe het programma gebruikt moet worden. Mijn mening is dat deze music-composer betere resultaten geeft als de eerdere music-composers op de Commodore 64.

GeoMusicPlayer

Op deze diskette staat nog een applicatie, die direct met de hiervoor besproken G.O. Bach te maken heeft, en wel GeoMusicPlayer. Deze applicatie is speciaal ontwikkeld om composities, die zijn gemaakt met G.O. Bach, af te kunnen spelen. Als men het programma heeft opgestart, verschijnt een bedieningspaneel van een CD-speler. U kunt dan een file, die is gemaakt met G.O. Bach inladen en laten afspelen. Alles zit er op, pauzestand en versneld vooruit en achteruit spoelen.

Spellen

Verder bevat de GEOS Companion diskette een tweetal spelletjes, op zich niet zo uniek, maar toch de moeite waard. De eerste is het spel Decode!, een MasterMind achtig spelletje, waarbij u de code moet doorbreken. De moeilijkheidsgraad is door u zelf te bepalen. Netjes uitgevoerd in kleuren en met een muziekje op de achtergrond. Het tweede spel is een verbeterde versie van Breakout, alleen deze is in een drie-dimensionale versie. Het vereist wel enige behendigheid om dit spel te spelen, omdat het balletje een natuurlijke beweging maakt en daardoor vrolijk in het rond stuitert. Bij 5000 punten krijgt u een extra bal, terwijl elk blok wat wordt weggetikt 100 punten oplevert.

GeoAnimator

Verreweg de mooiste applicatie op deze diskette is wel het programma GeoAnimator. Deze applicatie stelt u in staat om stand-alone animaties te maken, die zowel draaien onder GEOS 64 als wel onder GEOS 128. U maakt een animatie door een tekening te maken in GeoPaint en deze naar een fotoalbum te copiëren. Met GeoWrite schrijft u het script voor de animatie. Hierbij maakt u gebruik van de in de handleiding vermelde graphics instructies. Eigenlijk zijn deze instructies een soort programmeertaaltje op zich. Het script dat door het programma wordt gebruikt bouwt in feite de animatie op uit een aantal commando's, die soms een of meer argumenten nodig hebben. Een duidelijke uitleg in de handleiding begint met het verklaren hoe de graphics in GEOS zijn geregeld. Indien u ervaren bent met Basic programmeren, dan heeft dit programma geen geheimen voor u en kunt u gelijk aan de slag. Hiermee bedoel ik niet dat onervaren programmeurs het

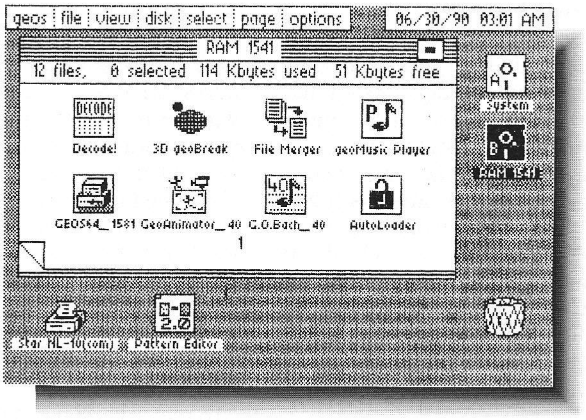
programma niet kunnen gebruiken. Het programma is zeer gebruikersvriendelijk en het resultaat is verbluffend. Bewegende plaatjes op uw scherm, met de resolutie, zoals u van GEOS gewend bent. En om het geheel te completeren, kunt u ook nog een muziekje op de achtergrond laten horen, die is vervaardigd met G.O. Bach. Kortom een compleet editorsysteem voor animaties in GEOS.

Utilities

Naast deze vier forse applicaties bevat de diskette nog een aantal nuttige GEOS utilities, zoals een nieuwe versie van de Pattern Editor, die nu ook geschikt is voor de 128-versie. Verder een applicatie FileMerge, waarmee GeoWrite files aan elkaar gekoppeld kunnen worden. Voor de 1581 diskdrive gebruikers is de applicatie 1581 Boot-Disk maker een uitkomst. Het programma creeërt bootable 1581-format diskettes. Het programma schrijft zijn eigen GEOS en GEOS Boot files naar de diskette en copieert tevens de GEOS Kernal van uw bestaande diskette. Het laatste programma is de applicatie Batch Copier, waarmee u een lijst van te copiëren bestanden kunt opgeven. En als laatste bevat deze diskette weer een aantal nieuwe letterfonts en clip-art, welke zijn vervaardigd door Susan Lamb, inmiddels een fenomeen op het gebied van GEOS-applicaties. De diskette is verkrijgbaar bij de redactie van Run Magazine uit Amerika en kost met verzendkosten zo'n $ 29,00, wat neer komt op een 55 gulden (bij de huidige koers). Wilt u meer informatie stuur dan een briefje naar IDG Communications, 80 Elm Street, Peterborough, NH 03458, USA.

Meer nieuws

Dat in Amerika meer wordt gedaan met het besturingssysteem GEOS is inmiddels wel duidelijk. Terwijl in ons land Commodore 64 en GEOS dreigen te verdwijnen in de kelderkast, worden in Amerika nog steeds noviteiten voor deze vertrouwde 8-bit computer ontwikkeld. Met name op het gebied van GEOS zijn er weer een tweetal goed bruikbare dingen bij gekomen.



Figuur 2

GeoRam 512

Voor een ieder die altijd al vond dat een GEOS scherm er te lang over deed, voordat het weer zichtbaar was op de monitor of dat een applicatie er te lang over deed om geladen te worden, is er dan eindelijk goed nieuws. Berkeley Softworks is dan eindelijk met de RAM expansie, van maar liefst 512 Kb, op de markt gekomen.
Zeker gezien de schaarste van Commodore RAM-expansie units is dit een welkome oplossing. De snelheid waarmee uw GEOS programma's nu werken is maar liefst 35 maal sneller dan een 1541, 1571 of 1581 diskdrive. De GeoRam expansie card wordt tevens geleverd met een upgrade van GEOS 2.0, die geschikt is voor gebruik op de 64 en 128. Helaas is deze card nog niet in Nederland te koop en in Amerika alleen rechtstreeks bij Berkeley Softworks voor een prijs van $ 129,90.

GeoBasic

Ja u leest het goed, GeoBasic. Net wanneer de meesten onder u misschien dachten dat het afgelopen zou zijn met ontwikkelingen onder het GEOS besturingssysteem komt men weer met iets nieuws. Nou ja, nieuw is het niet. Al vaker werd in met name Amerikaanse software magazines geadverteerd met GeoBasic. Alleen werd dit pakket nooit geleverd. Een alternatief was dan meestal Becker Basic van Abacus. Maar ook hierover heb ik in het verleden weinig gehoord. Nu is het mogelijk om een point-en-klik besturing aan uw eigen basic programma's te geven. Door Berkeley Softworks ontwikkeld en door Run magazine gedistribueerd, komt GeoBasic voor een prijs van $ 39,95 op de markt. Het is de langverwachte high level programmeertaal voor de C64/C128 GEOS-gebruiker. Met GeoBasic, dat meer dan 100 nieuwe opdrachten kent, kunt u al uw programma's voorzien van het professionele uiterlijk en gevoel, zoals pulldown menu's, iconen, dialoogvensters en veel meer. Het programma bestaat uit maar liefst 5 editors, een menu-editor, een bitmap-editor, een editor voor iconen en dialoogvensters en natuurlijk een sprite-editor. Het programma wordt geleverd met verschillende voorbeeld-programma's, zodat u gelijk aan de slag kunt. Wilt u meer informatie, schrijf dan een briefje naar Run magazine. Het adres staat eerder in deze GEOS-Info.

Van Basic naar GeoWrite

En alvast vooruitlopend op een mogelijke aanschaf van GeoBasic kunt u met onderstaande listing en bijbehorende beschrijving uw Basic 2.0 of 7.0 listing converteren naar GeoWrite. Laadt eerst uw Basic programma en voer dan de volgende opdrachten in de zogenaamde direct-modes in:

```
OPEN 2,8,2,
"O:bestandsnaam,S,W"
CMD 2:LIST:
PRINT#2:CLOSE2
```

Nadat het lampje van de diskdrive uit is, staat het programma op diskette. Start nu GEOS op en laadt het programma TextGrabber. U selecteert, afhankelijk van het gebruikte systeem, C128 GenericII of C64 GenericII. Vervolgens voert u een filenaam in voor het bestand. Plaats een diskette in de drive en de TextGrabber converteert de file naar GeoWrite formaat.

Bert Venema

TIPS & TRUCS 64

Terwijl u heerlijk lag te bakken in de Spaanse zon aan de Costa del Sol, had uw Commodore 64 eindelijk rust. Veilig lag hij in een donker hoekje in de kast, na een jaar lang afgeranseld te zijn door wanhopige programmeurs en geflipte video game freaks. Maar nu de vakantie voorbij is, wordt zijn rust ruw verstoord, want het is weer tijd voor Tips & Trucs.

Bij het aanslaan van sommige toetsen verschijnen er karakters op het scherm, bij andere toetsen verandert de positie van de cursor en bij weer andere toetsen verandert de kleur van de cursor. Om bij elke toetsaanslag de juiste handeling uit te voeren, is er een programma nodig, dat de invoer van het toetsenbord analyseert en op grond van de uitkomst van de analyse een bepaalde handeling uitvoert. Dit programma wordt de screen editor genoemd. De screen editor bevindt zich in het operating system van de Commodore 64.
Om correct te kunnen functioneren, moet de screen editor een aantal verschillende zaken bijhouden. Een aantal van deze zaken staan hieronder opgesomd:

-positie van de cursor
-kleur van de cursor
-knipperen van de cursor
-repeteren van bepaalde toetsen
-editor modes

Over de eerste vier punten hebben we het in een eerdere aflevering van 'Tips & Trucs 64' al gehad. In deze aflevering willen we aandacht besteden aan het laatste punt, de editor modes.
De screen editor kan zich in een aantal verschillende modes (toestanden) bevinden. Deze modes zijn: quote mode, insert mode en reverse mode. Voor elk van deze modes is er een adres beschikbaar, waarin wordt bijgehouden of de editor zich al dan niet in die mode bevindt. Hieronder zullen we het adres van elke editor mode geven met een beschrijving van de mode.

212: Quote mode

Als adres 212 een waarde ongelijk aan 0 bevat, dan bevindt de screen editor zich in de quote mode. De quote mode kan worden aangeschakeld door het quote-teken (") in te tikken.
Bij de aanslag van een aantal toetsen, zoals de cursor- en de HOME-toets, wordt er normaal geen karakter afgedrukt, maar wordt de positie van de cursor veranderd. Als de screen editor zich echter in de quote mode bevindt, dan wordt er bij de aanslag van deze toetsen wel een karakter afgedrukt en wordt de handeling, zoals het verplaatsen van de cursor, niet uitgevoerd. Een uitzondering hierop is de DELETE-toets. Deze toets blijft in de quote mode gewoon functioneren.
De quote mode kan worden uitgeschakeld door nog een keer het quote-teken in te tikken of door het aanslaan van de RETURN-toets.

216: Insert mode

Een waarde groter dan 0 geeft aan dat de screen editor zich in de insert mode bevindt. De insert mode wordt aangezet door op de INST-toets te drukken. Elke keer dat op de INST-toets wordt gedrukt, wordt de waarde in adres 216 verhoogd. Deze waarde is dus gelijk aan het aantal karakters dat nog ingevoegd kan worden.
De screen editor werkt in insert mode net zoals in quote mode met het verschil dat de DELETE-toets ook niet meer functioneert.
Bij het aanslaan van een toets anders dan de INST-toets, wordt de waarde van adres 216 met 1 verlaagd. De insert mode wordt dus uitgeschakeld als er genoeg karakters zijn ingevoegd. Ook wordt de insert mode uitgeschakeld bij het aanslaan van de RETURN-toets.

199: Reverse mode

Een waarde ongelijk aan 0 betekent dat de screen editor zich in de reverse mode bevindt. Door tegelijkertijd de CTRL-toets en de 9-toets (RVS ON) in te drukken wordt de editor in reverse mode gebracht.
In reverse mode wordt elk karakter dia-negatief afgebeeld.
De reverse mode kan weer worden uitgezet door de CTRL-toets en de 0-toets (RVS OFF) in te drukken of door op de RETURN-toets te drukken.

Soms is het handig om de verschillende editor modes uit te kunnen schakelen, zonder dat er iets op het scherm gebeurt, doordat u de quote- of RETURN-toets moet indrukken. U wilt bijvoorbeeld de insert mode uitschakelen, omdat u de DELETE-toets wilt gebruiken. Het volgende programma stelt u in staat om de drie verschillende editor modes uit te schakelen door tegelijkertijd op de CTRL-toets en de F1-toets te drukken. Het programma is een machinetaal programma dat in de interrupt routine loopt. Zodra het programma ziet dat de CTRL- en de F1-toets zijn ingedrukt, wordt de waarde 0 in elk van drie adressen voor de editor modes gezet, waardoor deze modes worden uitgezet.

```
5   rem uitschakelen
    editor modes
10  for i= 49152 to
    49188
20  read a:s=s+a:poke
    i,a
30  next
40  if s=4503 then
    sys 49152:end
50   print"fout in
    data!"
400 data
    120,169,13,141,20,
    3,169,192
410 data
    141,21,3,88,96,165
    ,145,201
420 data
    251,208,14,165,203
    ,201,4,208
430 data
    8,169,0,133,212,13
    3,199,133
440 data
    216,76,49,234,0
```

Na het indrukken van STOP/RESTORE moet het programma opnieuw worden geactiveerd met SYS 49152.

Toetsenbord scannen

Bij randapparatuur denkt u waarschijnlijk aan apparaten die van de computer losgekoppeld kunnen worden, zoals de disc-drive, printer en joysticks. Ook het toetsenbord behoort tot de randappa-

	ADRES 56321							
ADRES 56320	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
bit 7	STOP	Q	C=	SPACE	2	CTRL	←	1
bit 6	/	↑	=	RIGHT SHIFT	HOME	;	*	£
bit 5	,	@	:	.	-	L	P	+
bit 4	N	O	K	M	0	J	I	9
bit 3	V	U	H	B	8	G	Y	7
bit 2	X	T	F	C	6	D	R	5
bit 1	LEFT SHIFT	E	S	Z	4	A	W	3
bit 0	CRSR DOWN	F5	F3	F1	F7	CRSR RIGHT	RETURN	DELETE

Figuur 1

ratuur. Dit klinkt misschien raar, omdat het toetsenbord onmisbaar is voor het werken met de computer. Hoe zou u immers instructies aan de computer moeten geven zonder toetsenbord? De signalen die door het toetsenbord worden afgegeven, kunnen echter niet direct door de 6510 microprocessor verwerkt worden. Het toetsenbord is aangesloten op de Complex Interface Adapter #1 (CIA #1). Deze chip regelt de communicatie met de randapparatuur. Daarom wordt het toetsenbord als randapparaat beschouwd.

De 6510 microprocessor kan met de CIA communiceren om gegevens van een randapparaat te ontvangen of naar een randapparaat te zenden. De communicatie tussen de CIA en 6510 vindt plaats via de adressen 56320-56335 ($DC00-$DC0F).

Gegevens die van het toetsenbord afkomstig zijn, kunnen via Data Port A en B van de CIA gelezen worden. De adressen van deze twee poorten zijn respectievelijk 56320 en 56321. De acht bits van deze poorten kunnen zowel als input als output bits gebruikt worden. Als een bit als input bit gebruikt wordt, dan kan er via dit bit een 0 of een 1 van de CIA ontvangen worden; via een output bit kan een 0 of 1 naar de CIA gestuurd worden. Om het toetsenbord te lezen, moeten alle bits van poort A output bits zijn en alle bits van poort B input bits.

De inhoud van de adressen 56322 en 56323 bepalen of een de bits van Data Port A en B input of output bits zijn. Deze twee adressen worden Data Direction Port A en B genoemd. Als bit 0 van adres 56322 gelijk is aan 0, dan is bit 0 van adres 56320 een input bit en anders een output bit. Zo correspondeert elk bit van adres 56322 met een bit van adres 56320. Op dezelfde wijze corresponderen de bits van adres 56323 met de bits van adres 56321. Met POKE 56322,255 en POKE 56323,0 wordt er dus voor gezorgd dat alle bits van Port A output zijn en alle bits van Port B input. Bij het aanzetten van de computer wordt dit automatisch gedaan.

De toetsen van het toetsenbord zijn via een matrix verbonden met de CIA. Deze matrix is in figuur 1 weergegeven. Met behulp van deze matrix kan nu als volgt bepaald worden welke toets er is ingedrukt. Eerst moet er een rij van de matrix geselecteerd worden. Vervolgens kan bepaald worden welke toetsen in deze rij zijn ingedrukt.

Elk bit van adres 56320 correspondeert met een rij uit de matrix (zie figuur 1). Door een bit op 0 te zetten en de andere bits op 1 wordt de rij, corresponderend met het bit dat op 0 is gezet, geselecteerd. Nu kan via adres 56321 bepaald worden welke toetsen er in deze rij zijn ingedrukt. De bits van adres 56321 corresponderen met de kolommen van de matrix. Een bit in dit adres krijgt de waarde 0 als de corresponderende toets in de geselecteerde rij is ingedrukt.

We geven een voorbeeldje. U zet bit 2 van adres 56320 op 0 en de rest op 1. Vervolgens leest u de waarde van adres 56321. U ontdekt dat bit 5 van deze waarde op 0 staat. Dit betekent dat de letter 'F' is ingedrukt.

Om nu te bepalen welke toetsen er zijn ingedrukt, moet u elke rij een keer selecteren en per rij kijken welke toetsen er zijn ingedrukt. Dit is natuurlijk een nogal omslachtige manier. Normaal doet de interrupt routine dit werk voor u. Deze scant 60 keer per seconde het toetsenbord. U kunt dan bijvoorbeeld via adres 203 bepalen welke toets er is ingedrukt. Als dit adres de waarde 64 heeft, dan is er geen toets ingedrukt. Een waarde kleiner dan 64 betekent dat er een toets is ingedrukt. Elke waarde komt overeen met een bepaalde toets. De toetsen zijn daarbij als volgt genummerd: 8 * rij bit + kolom bit. Rij bit en kolom bit zijn de bitnummers die in figuur 1 bij de rijen en kolommen van de matrix staan. De letter 'F' heeft zo de waarde 8*2+5=21. Op

deze manier is het echter niet mogelijk om te testen of er meerdere toetsen tegelijk zijn ingedrukt. Daarvoor moeten wel de adressen 56320 en 56321 gebruikt worden. Het programma in listing 1 geeft hiervan een voorbeeld. Het programma drukt de matrix van figuur 1 af op het scherm. De toetsen die zijn ingedrukt, worden in reverse weergegeven. Omdat het programma in Basic is geschreven, is het helaas nogal traag. Nadat u een toets heeft ingedrukt, kan het enkele seconden duren voordat dit op het scherm zichtbaar is. Dit kan verholpen worden door het programma in machinetaal te herschrijven. We hebben echter voor een Basic programma gekozen, omdat dit de methode beter verduidelijkt.
In regel 20 wordt de interrupt uitgeschakeld. Dit is nodig, omdat de interrupt 60 keer per seconde het toetsenbord scant. Tijdens dit scannen wordt de waarde van adres 56320 veranderd, waardoor het onmogelijk is om zelf een waarde in dit adres te poken. In regel 40 worden achtereenvolgens rij 0 tot 7 van de matrix geselecteerd. In regel 50 wordt de waarde van adres 56321 aan de variabele 'k' toegekend. In regel 80 wordt getest welke bits van deze waarde op 0 staan.
Zorg ervoor dat u het programma eerst op tape of disk zet, voodat u het runt. Als het programma eenmaal gestart is, kan het niet meer gestopt worden, omdat de interrupt is uitgeschakeld.

Muterende programma's

Vroeger, zeg zo'n 25 jaar geleden, waren zelf-muterende programma's populair. Muterende programma's zijn programma's die veranderen tijdens het uitvoeren van het programma, waardoor er steeds andere dingen worden gedaan. Tegenwoordig wordt het maken van dergelijke programma's beschouwd als slecht en ongestructureerd. Een zelf-muterend programma wordt dan ook meteen onder de noemer 'smerig programmeren' gezet. Niet ten onrechte, want zulke programma's zijn vaak slecht leesbaar. Toch zullen we hier verder ingaan op muterende programma's, omdat er vaak snelheid en geheugen winst behaald kan worden; twee zeer belangrijke punten voor de 64-programmeur.
Zoals gezegd is een zelf-muterend programma, een programma dat de code van het programma zelf verandert tijdens de uitvoer ervan. Dit betekent dat tijdens de uitvoer van het programma in het geheugen geknoeid moet worden, om het programma te veranderen. Dit brengt met zich mee, dat de programmeur heel precies moet weten waar elke instuctie precies in het geheugen staat. Bij Basic programma's is dit vrijwel uitgesloten. Bij het programmeren in machinecode gaat het echter beter. Als er in een machinetaal monitor geprogrammeerd wordt, ziet de programmeur meteen waar alle machinetaal instructies in het geheugen staan. Het is dan vrij makkelijk om dan zelf-muterende programma's te schrijven. Om het begrip 'zelf-muterend' duidelijker te maken geven we een voorbeeld. We maken eerst een programma die het hele beeldscherm vol zet met de schermcode $FF:

```
c000 lda #$00
c002 sta $fb
c004 lda #$04
c006 sta $fc
c008 lda #$ff
c00a ldx #$00
c00c ldy #$00
c00e sta ($fb),y
c010 iny
c011 cpy #$00
c013 bne $c00e
c015 inc $fc
c017 inx
c018 cpx #$04
c01a bne $c00c
c01c rts
```

Het bovenstaande programma zet het beeldscherm vol met $FF op de conventionele manier. De opbouw van het programma spreekt voor zich. We kunnen dit zelfde effect ook op een snellere en kortere manier verkrijgen, als we een zelf-muterend programma schrijven:

```
c000 lda #$ff
c002 ldy #$00
c004 ldx #$00
c006 sta $0400
c009 inx
c00a inc $c007
c00d cpx #$00
c00f bne $c006
c011 iny
c012 inc $c008
c015 cpy #$04
c017 bne $c004
c019 rts
```

Als u naar dit programma kijkt, ziet u dat in de regels $cooa en $co12 de loop van het programma beinvloed wordt, door rechtstreeks in de code te gaan poken. In regel c006 wordt er $FF op het scherm gepoked. Door in de code te poken, wordt het schermadres in regel $c006 steeds verhoogd, totdat het hele scherm volstaat met $FF.
Let wel, bij een tweede aanroep van dit stukje code, wordt niet het scherm volgezet met $FF, maar de eerste 4K geheugen achter het schermgeheugen. Dit is het grootste nadeel van zelf-muterende programma's. Bij het meerdere malen uitvoeren van het programma, worden er steeds andere resultaten verkregen. De meest gebruikte toepassing van zelf-muterende programma's op de Commodore 64 is het verplaatsen van blokken geheugen. we geven hier geen voorbeeld programma van, maar laten het schrijven van een dergelijk programma over aan de lezer.

Overbodig gebruik INT

De instructie INT maakt van een breuk (een getal met cijfers achter de komma, oftewel floating point getal) een geheel getal. Daarbij rondt INT (INT staat voor integer: geheel getal) elk getal naar beneden af, zodat er een geheel getal overblijft. Dit is dus niet hetzelfde als afronden, omdat int(9.9) het getal 9 oplevert, en niet 10. Om af te ronden moet bij het getal in kwestie nog een 1/2 opgeteld worden:

```
print int(getal + .5)
```

Het bovenstaande statement zal de variabele 'getal' afronden en uitprinten.
Er wordt echter vaak overbodig gebruik gemaakt van de instructie INT. Bij een heleboel Basic instructies wordt het meegegeven getal automatisch geconverteerd naar een geheel getal. Het is dan dus overbodig om dit eerst zelf te doen met INT. Als voorbeeld noemen we de instuctie POKE.

```
POKE 53280, 0.999
```

Als het bovenstaande ingevoerd wordt, wordt gewoon de insructie POKE 53280,0 uitgevoerd. We geven hieronder een lijst met instructies, waarbij U de INT instructie niet hoeft te gebruiken.

```
CHR$
DIM
LEFT$
MID$
PEEK
POKE
RIGHT$
SPC
SYS
TAB
```

Aanwezigheid disk-drive of printer

Soms is het handig om te weten of de disk-drive aanstaat. U wilt bijvoorbeeld in een programma een hires-plaatje inladen. Daarvoor wilt u eerst testen of de disk-drive wel aanstaat. Immers, als u gewoon begint met laden en de disk-drive staat nier aan, dan wordt er een foutmelding gegeven, en stopt de program-

ma. Om dit te vermijden, moet er eerst even getest worden of de disk-drive wel aanstaat. Dit kunt u doen met het volgende programma:

```
10  open 2,8,2
20  close 2
30  if st = -128 then
    print"drive not
    ready":end
40  print"drive
    ready"
```

Eerst worden er een OPEN en een CLOSE gedaan. Als daarna ST (Status variabele) gelijk is aan -128, staat de disk-drive niet aan. Als ST gelijk is aan 0, dan staat de disk-drive wel aan.
We kunnen eenzelfde programma schrijven voor de printer. Hier komt het programma, dat test of de printer aanstaat:

```
10  open 1,4,7
20  close 1
30  if st = -128 then
    print"printer
    not ready":end
40  print"printer
    ready"
```

Het programma is analoog aan het eerste programma voor de disk-drive.

Merge voor de cassette

Het komt wel eens voor dat u twee programma's heeft, die u achter elkaar wilt zetten. U kunt dan niet de twee programma's gewoon achter elkaar inladen, omdat u dan in de knoop komt mt Basic-pointers. Hieronder beschrijven we een manier hoe u twee programma's op tape, achter elkaar kunt zetten.

- Als eerste moet u er opletten dat de twee programma's verschillende regelnummers hebben. Het tweede programma moet verder hogere regelnummers hebben, dan het eerste programma, zodat het tweede programma precies achter het eerste programma komt te staan.
- Laadt nu het tweede programma in.
- Typ het volgende in:

```
OPEN 1,1,1,"prognaam
    2":CMD 1:LIST
```

Het tweede programma wordt nu sequentieel op tape gesaved.
- Typ in: PRINT#1:CLOSE1

- Laadt nu het eerste programma in.
- Laadt nu het tweede, sequentieel gesavede, programma weer in met:

```
POKE 19,1:OPEN 1,1,
    0,"prognaam 2"
```

- Als de computer FOUND geeft, moet u op de Commodore-logo toets drukken. Voer nu de volgende handelingen uit:
- Maak het scherm schoon met CLR/HOME en druk drie keer op CRSR DOWN. Typ nu het volgende in:

```
POKE 153,1:POKE198,
    1:POKE 631,13:
    PRINTCHR$(13)
```

ALs de computer nu klaar is met laden, staan de twee programma's gemerged in het geheugen.

Peeks & Pokes:

61 en 62:
In de adressen 61 en 62 staat het beginadres van de regel opgeslagen, waar de computer verdergaat met uitvoeren, als er CONT wordt ingevoerd. Het lo- en hi byte van dit adres is opgeslagen in respectievelijk adres 61 en adres 62. Als er na het stoppen van het programma iets veranderd is aan het programma, of als er een foutmelding is gegeven, dan staat in adres 61 de waarde 10 en in adres 62 de waarde 0. Bij een CONT wordt dan de foutmelding "Can't continue error" gegeven.

63 en 64:
In de adressen 63 en 64 staat het huidige data-regel nummer. Dit betekent dat de computer in deze twee adressen bijhoudt van welke Basic-regel het volgende data-getal gelezen moet worden bij een READ. Daarbij staat in adres 63 het low- en in adres 64 het hi-byte van het regelnummer. Als de computer bijvoorbeeld stopt met een 'Illegal quantity error' bij een READ, kan met behulp van deze adressen gekeken worden, bij welke data-regel de computer was met lezen:

```
PRINT PEEK(63) +
    256*PEEK(64)
```

19:
In adres 19 wordt het huidige I/O apparaat bijgehouden. Daarbij heeft het beeldscherm de code 0. Dit betekent dat alle beeldscherm meldingen worden onderdrukt als er geen 0 in dit adres staat. Zo wordt er bijvoorbeeld geen vraagteken afgedrukt bij een INPUT als er geen 0 in dit adres staat.

Michel de Boer & Hylke Spranger

LISTING 1

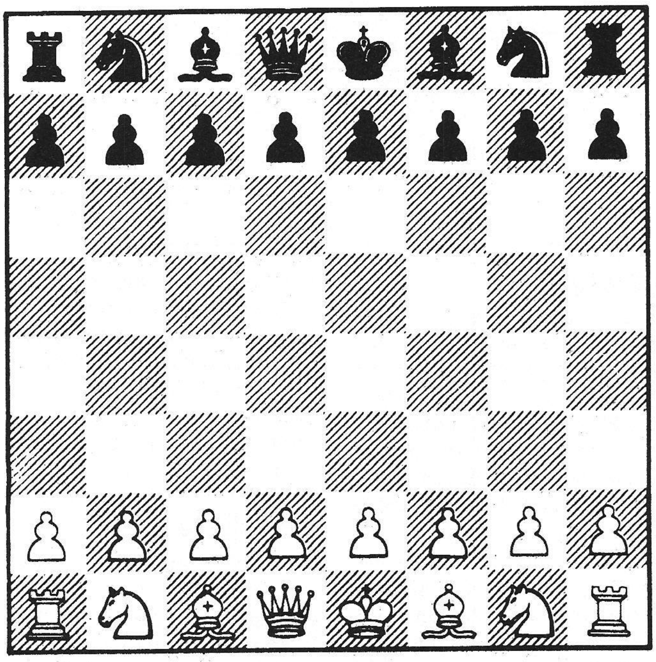
```
5   rem ******************
10  rem * keyboard matrix *
15  rem ******************
16  rem
20  poke 56334,0
30  print"[SHIFT-CLR]";:gosub 120
40  for i=0 to 7:poke 56320,255-2^i
50  k=peek(56321):x=1026+(7-i)*120+7*5
60  for j=0 to 7
70  poke x,peek(x) and 127
80  if (k and 2^j)=0then poke x,peek(x) or 128
90  x=x-5
100 next j,i
110 goto 40
120 print" stp   q    c=   spc   2   ctl          1":print:print
130 print"  /    ^    =    rsh  hom   ;     *     |":print:print
140 print"  ,    `    :     .    -    l     p     +":print:print
150 print"  n    o    k     m    0    j     i    9":print:print
160 print"  v    u    h     b    8    g     y     7":print:print
170 print"  x    t    f     c    6    d     r     5":print:print
180 print" lsh   e    s     z    4    a     w     3":print:print
190 print" crd   f5   f3    f1   f7  crr   ret   del";
200 return
```

The Final Chesscard

Een schaakcartridge voor de 64

Het computerschaken is al zo oud als de computer zelf. Aan het eind van de veertiger jaren werd de eerste computer ontwikkeld. Het duurde daarna niet lang meer voor het eerste schaakprogramma gemaakt werd. Opvallend feit daarbij is, dat de grondlegger van de computer, John von Neumann, ook aan de basis van het computerschaken heeft gestaan, samen met Claude Shannon en Alan Turing. Met de verbetering van de computers werden ook de schaakprogramma's beter en beter. Chess 4.0, Belle en Deep Thought zijn daarbij grote namen in de computerschaak geschiedenis. Ook voor de Commodore 64 zijn er een aantal schaakprogramma's gemaakt, zoals Grandmaster, Chessmaster en Colossus. Sedert enkele tijd is er nu een schaakcartridge op de markt: The Final Chesscard, met een elo-rating van maar liefst 2000. Wij testten de Commodore 64 versie van deze cartridge.

De Final Cartridge wordt in twee uitvoeringen geleverd: een Commodore 64 en een PC versie. De PC versie heeft een speelsterkte van 2100 Elo en wordt onder meer door Hans Boehm gebruikt bij zijn tv-presentaties. In de rest van dit artikel richten we ons echter op de Commodore 64 versie van deze schaakcartridge.



De cartridge, die groot van formaat is, wordt in een kleurige doos geleverd. Bij de cartridge wordt nog een duidelijke handleiding geleverd. Voor dit alles moet u 199 gulden neertellen.

De Cartridge opstarten

Als u de Final Chesscard opstart, krijgt u gelijk een schaakbord op het beeld, dat bijna het hele scherm vult. Rechtsboven op het scherm bevinden zich twee klokken, die de schaaktijd van respectievelijk zwart en wit bijhouden. Daaronder wordt de partijnotatie automatisch bijgehouden. De cartridge kan bestuurd worden met het toetsenbord, een joystick of een muis. Wij testten de cartridge met een joystick.

Als de cartridge is opgestart kan er gelijk begonnen worden met schaken. Dit gebeurt door met de joystick een stuk aan te wijzen en te verplaatsen. Als de speler een zet heeft gedaan, doet de computer automatisch een tegenzet. Het gebruik van de joystick is zeer eenvoudig. U ziet aan een kruisje op het scherm waar u bent met de joystick. Dit kruisje verdwijnt na een tijdje, als u de joystick niet aanraakt. Met behulp van de vuurknop kunt u de stukken dan verplaatsen
Verder is de cartridge zeer gebruikersvriendelijk. In het begin komt de handleiding er nauwelijks aan te pas. Slechts als we wat dieper in willen gaan op alle mogelijkheden die de cartridge biedt, hebben we de handleiding nodig.

Aan de rechterkant van het scherm staan ook nog twee pijltjes. Als met de joystick op het linker pijltje wordt gedrukt, neemt de computer een zet terug. Met het rechter pijltje kan er dan eventueel weer een zet vooruit worden gedaan. We vinden het wel jammer dat de schaaktijd niet wordt gecorrigeerd als er een zet terug wordt gedaan: de tijd wordt niet teruggezet, maar blijft gewoon doorlopen.

Mogelijkheden

Hoewel er meteen begonnen kan worden met schaken, heeft de cartridge ook een heleboel opties die ingesteld kunnen worden. Boven aan het scherm bevindt zich een menubar met pulldown menu's (zie figuur 1). Deze menubar doet direct aan Geos denken, en is onderverdeeld in de volgende rubrieken:

- Options
- Board
- Settings
- Disk
- Features

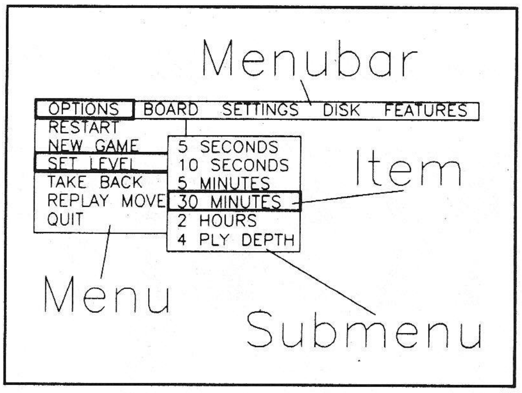
Onder elk van deze 5 namen, bevinden zich een (groot) aantal funkties waarmee het gedrag en uiterlijk van de card beinvloed resp. bepaald kunnen worden. We zullen nu voor elk van de 5 rubrieken globaal aangeven welke functies ze omvatten. Elk van deze functies kan eenvoudig met de joystick geselecteerd worden, net zoals bij Geos.

Options

De rubriek Options omvat de meeste functies. We zullen de belangrijkste hiervan bespreken.
De functie Restart spreekt voor zich: hier kan een nieuw spel mee worden begonnen.
Met de functie Preset Level kunt u instellen hoeveel tijd de spelers hebben en hoe lang de gemiddelde tijd per zet is. U kunt bijvoorbeeld instellen dat de computer oneindig veel tijd heeft met denken. Op deze manier kan de computer een stelling (bijvoorbeeld een eindspel) volledig analyseren. Als u echter vindt dat de computer lang genoeg heeft nagedacht en een zet moet doen, kunt u de functie Stop Thinking selecteren; de computer zal dan gelijk een zet doen.
Met de functie Play Mode kunt u uitkiezen wie tegen wie speelt. U kunt bijvoorbeeld de computer tegen zichzelf laten spelen.
Verder zit er nog een functie Take Back All in, die alle gedane zetten terugneemt. Door daarna de functie Replay All te gebruiken, worden alle zetten weer vooruit gespeeld.
Onder deze rubriek misten we een Undo functie, waarmee de laatst gedane handeling teniet kan worden gedaan. Het kan bijvoorbeeld voorkomen dat u midden in een spel per ongeluk op Restart drukt. U bent dan de hele stelling gelijk kwijt.

Board

Als u de functie Teaching gebruikt, geeft de Chesscard steeds wanneer u een zet doet, alle velden aan op het bord waar het uitgekozen stuk heen kan.
Als u aan zet bent, kunt u met Suggest Move aan de Final Chesscard de zet opvragen die hij gespeeld zou hebben.
Met de functies Menu Colors, Piece Colors en Screen Colors kunnen de kleuren ingesteld worden van het menu, de speelstukken, en het scherm. Verder kan de snelheid waarmee de stukken bewegen ingesteld worden.
We misten onder deze rubriek een functie waarmee de snelheid van het aanwijskruisje ingesteld kon worden. We vinden namelijk dat het kruisje net iets te traag over het scherm loopt.



Settings

Met de functie Program kunt u het speelgedrag van de card instellen. U kunt aangeven of de card met of zonder openingenboek speelt. Ook kunt u de cartridge in de beginnersmode zetten, de cartridge zal dan minder goed spelen.
Met de optie Brute Force kunt u ook nog het algoritme instellen dat de computer gebruikt bij het schaken. Het gaat tever om in dit artikel precies aan te geven hoe de speelstijl allemaal veranderd kan worden.

Disk

Onder deze rubriek bevinden zich allemaal functies om stellingen te saven en te laden. De Final Chesscard heeft een ingebouwde RAM-chip die gevoed wordt door twee penlight batterijen in de cartridge. U kunt dan ook kiezen of u de stellingen wilt saven op deze RAM-disk, of op de gewone disk-drive. Verder kunt u ook de Settings en Board-opties die u allemaal heeft ingesteld saven en laden.

Features

De functie SetUp A Position stelt u in staat om zelf een stelling op het bord te zetten. U kunt met de joystick alle stukken op het bord zetten. U kunt daarbij nog van alles instellen, bijvoorbeeld welke kant de eerste zet moet doen.

Met de functie Solve For Mate kan de speciale matzoeker van de Final Chesscard geactiveerd worden. U kunt daarbij de zoekdiepte instellen. Solve For Mate is een handige functie bij het oplossen van schaakpuzzels.
Met de functie Print Board kunt u een stelling uit laten printen op de printer.

Conclusie

We hebben nu de belangrijkste functies van de Final Chesscard behandeld. We kunnen vaststellen dat de cartridge over zeer veel functies beschikt en dat we nauwelijks functies missen. Als extra mogelijkheid zit er ook nog een demo in, die laat zien hoe u om moet gaan met de cartridge. Verder is de cartridge zeer robuust geprogrammeerd: het is ons niet gelukt om de cartridge te laten crashen, iets wat niet van alle cartridges gezegd kan worden.
Ook willen we nog even de gebruikersvriendelijkheid van de Final Chesscard onderstrepen. Het is zeer plezierig om met de cartridge te werken en we zouden wensen dat elk stukje software zo gebruikersvriendelijk was als deze chesscard.
Rest er nog de prijs van 199 piek. Dit is niet een bedrag dat elke Commodore 64 bezitter even uit zijn of haar mouw schudt. Dit bedrag is mede zo hoog door de speciale processor (5 MHz!) die in de cartridge zit, en de lange ontwikkelingstijd (3 jaar). We kunnen dus concluderen dat als u een grote schaakliefhebber bent, dat u deze cartridge zeker moet kopen. U krijgt dan zeker waar voor u geld en u zult van goede huize moeten komen om deze Final Chesscard te verslaan.

Michel de Boer &
Hylke Sprangers

Specificaties:

Naam:	Final Chesscard
Makers:	TASV B.V.
Uitvoering:	Cartridge
Processor:	65C02, 5Mhz
Geheugen:	64 K ROM, 8 K RAM-disk met batterij-voeding
Speelsterkte:	2000 Elo
Computer:	C-64
Prijs:	*f* 199,-

PRINTERS EN GEOS

vragen en antwoorden......

Er bereiken mij de laatste tijd erg veel vragen over printers. Waarom zij wel of niet werken met GEOS en welke de beste is of er een interface nodig is of niet....

Enkele van deze vragen zal ik proberen te beantwoorden. Of u nu met de C64 of de C128 werkt maakt voor de antwoorden niet uit.

vraag:Wat moet je weten voordat een printer wordt aangeschaft om te kunnen werken met GEOS?

Een 80 dpi printer is op dit moment de beste koop die u kunt doen. Deze printer heeft grafisch de beste prijs/prestatie op dit moment. De voorwaarde is wel dat u een parallel/centronics uitvoering van deze printer koopt en niet de Commodore versie.
U dient wel een parallel/centronics kabel extra aan te schaffen.

vraag:Als ik een geoWrite dokument of geoPaint tekening wil printen verschijnt de melding "Printer is inaccessible" op het beeldscherm. Waarom?

Er zijn een aantal redenen waarom deze foutmelding op het scherm verschijnt. De meest voorkomende is dat een verkeerde printerdriver is gekozen. Probeer in de handleiding van uw printer te zoeken welke printers uw eigen printer kan simuleren en gebruik dan vervolgens de daarvoor bestemde printer. Ook kan het voorkomen dat u een centronics printer bezit en deze met een seriële printerdriver probeert aan te sturen. Voor centronics printers zijn aparte printerdrivers beschikbaar. Voorts wil het nog wel eens gebeuren dat de printer nog niet aan staat of nog niet op "on line" staat. Ook kan het zijn dat uw papier op is.

vraag:Wat betekend de melding" cannot find printerdriver" en wat kan ik hieraan doen om dit te vermijden?

Deze melding geeft aan dat u waarschijnlijk de verkeerde- of helemaal geen printerdriver naar uw werkdiskette heeft gekopieerd. Ook al staat de naam van de door u gekozen onder de printer op de desktop. De printerdriver hoort op de diskette te staan waar ook de dokumenten op staan. Ook als u uw applikatie op een andere disk heeft staan.

vraag:Waarom wordt mijn dokument wat ik wil printen helemaal verminkt?

Hier zijn verschillende redenen voor te noemen. bijv.: U heeft een printer die niet grafisch kan printen. Omdat in dit soort printers gebruik wordt gemaakt van de software die in de printer aanwezig is en niet van buitenaf gestuurd kan worden. Heeft u wel een printer die een grafische ondersteuning geeft dan is er waarschijnlijk weer een verkeerde printerdriver gekozen.
Gebruikt u bij uw printer een printer-interface dan zult u de interface in de transparant mode moeten zetten. Hoe u dit moet doen kunt u vinden in het instruktieboek dat met de interface werd meegeleverd.

vraag:Er zitten blanke lijnen tussen de letters, en met tekeningen printen is dat ook het geval. Wat is dit en hoe krijg ik ze weg?

Omdat iedere regel in Geos in principe 2 x geprint wordt met een kleine verschuiving van het papier kan het inderdaad voorkomen dat er een witte-blanke lijn verschijnt. Dit kunt u opheffen door de "Line-Feed" optie uit te schakelen. Dit gebeurd meestal door middel van een Dip-switch omzetten in de printer en/of interfacekaart. In het instruktieboek van uw printer en/of interface vind u welke dipswitch dat is. Zet deze op OFF en het probleem is ook OFF.

vraag:Kan ik met geoWrite mijn dokumenten in NLQ printen?
En zo ja, hoe moet ik dat dan doen?

Ja, dat is mogelijk met de geoWrite 2.1 versie. De voorwaarde is dat uw printer een NLQ mode heeft. U gaat als volgt te werk: eerst plaatst u het Commodore Font op uw werkdisk bij.
Vervolgens start u geoWrite en tikt u brief in het standaard BSW lettertype. Nadat u brief klaar is kunt u met C= V de hele pagina selekteren. De tekst staat nu in reverse op het scherm. U gaat met het pijltje naar het Font menu en kiest hier voor het Commodore 10 points Font. Vervolgens gaat u naar het "Page" menu en selekteer daar de optie NLQ-spacing. Dit zal voorkomen dat het dokument zonder spaties wordt geprint. Als u nu het dokument print zal deze softwarematig de printer in de NLQ stand zetten.
Er zijn printers waar geen softwareondersteuning mogelijk is. Voor deze printereigenaars de volgende tip:
Zet de printer hardwarematig in de NLQ stand. U hoeft niet het Commodore Font te selekteren. U kunt zo printen.
N.B.: Als u tekeningen in het dokument heeft staan worden deze natuurlijk niet geprint. Ook wordt met NLQ printen geen bold, underline of italics ondersteund.

vraag:Is GEOS compatibel met Daisy Wheel printers?

In principe niet. GEOS is een grafisch georiënteerd

programma en kan dan alleen door printers verwerkt worden die een grafische mode bezitten. Doch er is inmiddels een printerdriver voor dit soort printers ontworpen. ASCII Only is de toepasselijke naam van deze driver. In het print Optie menu van geoWrite kunt u met de keuze Draft en NLQ de printer aansturen. De optie High kunt u dan niet gebruiken. Eveneens als tekeningen printen met een Daisy Wheel printer is niet mogelijk.

vraag:Wat is een geoCable en wat doet hij?

Een geoCable is een parallel/centronics kabel die tussen de Userpoort van de computer en een centronics printer wordt aangesloten. Het voordeel van zo'n kabel is dat er een groot assortiment printers tot uw beschikking komt. Tevens is de overdracht veel sneller. Over de Commodore seriële bus wordt bit voor bit verstuurt en parallel worden er 8 bits tegelijk verstuurd. U heeft dan wel speciale geoCable printerdrivers nodig. U herkent deze printerdrivers aan hun toevoeging (gc).

vraag:Ik heb een MPS 1230 printer gekocht maar als ik deze centronics aanstuur print hij niet. Wel over de seriële poort. Hoe kan dat?

Als u een centronics printer wilt gebruiken voor GEOS dan zult u de speciale centronics printerdrivers moeten gebruiken. Deze staan helaas niet op de diskettes die door Berkeley worden geleverd. Een diskette met de centronics printerdrivers is verkrijgbaar bij de St. GEOS Gebruikers.

vraag:Als ik losse A4 vellen in mijn printer gebruik slaat mijn computer vast. Op het scherm blijft wel het dialoogvenster "printing" met daaronder CANCEL staan. Wat is er gebeurd?

In principe niets. Uw computer staat gewoon te wachten op input.
Maar waarschijnlijk is het de printer die inmiddels een paper out detectie heeft gedaan en verwacht een nieuw blad papier. Voert u nu een nieuw blad papier in dan zal het programma gewoon doorgaan en terugkeren naar de deskTop of de applikatie. Vergeet u niet de printer weer op "on line" te zetten.

vraag: Is GEOS compatibel met 24 naalds printers.

Ja. Er zijn twee 24 naalds drivers voor GEOS. De Star NB 15 driver staat op de systemdisk van GEOS 2.0. De Epson LQ1500 driver staat op de printerdisk van de St. GEOS Gebruikers.
De volgende printers zijn compatibel met deze drivers:
Epson LQ 1500, LQ 800, LQ 850, LQ 1000, LQ 1050, LQ 1500, LQ 2500, Seikosha SL-80AI, Star NB 15, Star NB 24-10 en de Star NB 24-15. Het is natuurlijk mogelijk dat er meer printers zijn die gebruik kunnen maken van deze drivers. Dit moet dan gewoon geprobeerd worden. Het is heel goed mogelijk dat door omschakeling van 1 dip-switch de printer wel werkt.

Tot slot

Zo, ik hoop dat een ieder hier de informatie uit kan halen die voor u van toepassing is en dan alsnog met GEOS van start kan gaan. Zijn er specifieke vragen over printers met GEOS dan zijn er mensen bij de St. GEOS Gebruikers die daar een heleboel van af weten. Er is elke dinsdagavond van 19:00 tot 22:00 uur een telefonisch spreekuur.

Pim Broekhuijzen

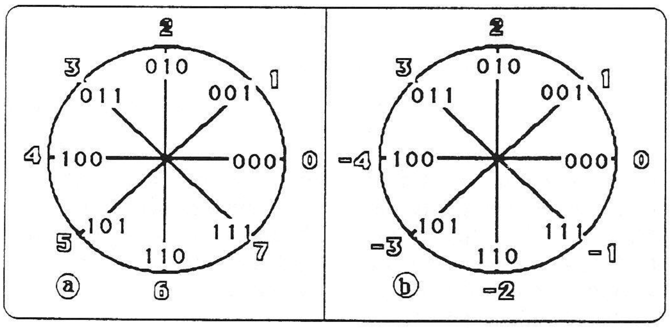
GEOS machinetaal (12)

rekenen en GEOS processen

Deze alweer twaalfde aflevering van de machinetaalcursus voor het GEOS operating system, is gewijd aan twee soorten nog niet eerder besproken routines: rekenroutines en procesroutines. Net zoals alle andere GEOS-functies zitten deze standaard in de GEOS kernal, en ze kunnen ten allen tijde door een applicatieprogramma worden aangeroepen. En zoals bekend: deze zeer uitgebriede routine- bibliotheek is nu juist dat element van GEOS dat het zo'n krachtig bedrijfssysteem maakt. De listing van de maand is het 'bouncing bars' programma, dit is gemaakt ter illustratie van de mogelijkheden van proces- en rekenroutines.

Getalsysteem

Een basisbegrip op het gebied van rekenen is het getalsysteem. De centrale vraag is: wat zijn getallen in GEOS? Ofwel: Hoe worden getallen gerepresenteerd? Allereerst zijn er twee soorten getallen: bytes en words. Een byte bestaat uit 8 bits en neemt dus 1 geheugenplaats in, en een word bestaat uit 16 bits en neemt dus twee geheugenplaatsen in. De eerste geheugenplaats (de eerste byte) heet de low-byte, en de daaropvolgende geheugenplaats heet de high-byte van het word. De waarde van het word is bepaalt als: lowbyte + high-byte*256. Dit is naar alle waarschijnlijkheid al bekende stof voor de vaste lezers van deze cursus.



Illustratie 1

Er is echter ook nog nieuws te melden. GEOS kent namelijk ook words die mogelijk negatief zijn. Dit soort words heten signed words (sign = teken). Words die per definitie alleen positief zijn, zoals in de alinea hierboven, heten unsigned. Voor het toestaan van negatieve getallen zijn helemaal geen extra bits nodig om aan te geven of het getal negatief is: er wordt gewoon een andere interpretatie (betekenis) van de bitpatronen afgesproken. Illustratie 1 verduidelijkt dit aan de hand van 3-bits getallen. In illustratie 1a staan alle mogelijke unsigned 3-bits getallen, in illustratie 1b alle mogelijke signed 3-bits getallen.

Twee-complementsysteem

Zoals U ziet geeft in 1b het hoogste bit wel het teken aan, namelijk als het hoogste bit 1 is, dan is het getal negatief, en als het 0 is, dan is het getal positief. Het is echter niet zo, dat bij het omdraaien van het hoogste bit de genegatieveerde versie van het oorspronkelijke getal verkregen wordt. De positieve getallen gaan normaal van 0 naar 3, en de negatieve getallen gaan dan van het alleronderste getal naar -1. Dit systeem wordt het twee-complement systeem genoemd. Het grote voordeel van dit systeem is dat optel en aftrekoperaties hetzelfde blijven als in unsigned gevallen. Kijk maar na, bijvoorbeeld: -2 + 1 = 110 + 001 = 111 = -1. Er kan dus gewoon van de optelling 110 + 001 gebruikgemaakt blijven worden. Het negatieveren van een getal in het twee-

GEOS REKEN ROUTINES

naam routine	adres	variabelen in	werking	variabelen uit
BBMult	$c160	x,y-ZeroPage addr	uBYTE(x) * uBYTE(y) in uWORD(x)	*(x) bevat produkt
Dnegate	$c172	x-ZeroPage adr	sWORD(x0 wordt -sWORD(x)	*(x) bevat -*(x)
Dabs	$c16f	x-ZeroPage addr	ABS(sWORD(x)) in sWORD(x)	*(x) bevat ABS *(x)
GetRandom	$c187	geen	trek 'n willekeurig getal(WORD)	random bevat getal
Ddec	$c175	x-ZeroPage addr	verminder een uWORD met 1	*(x) = *(x)-1
BMult	$c163	x,y-ZeroPage addr	uWORD(x) * uBYTE(y) in uWORD(x)	*(x) produkt,*(y)=0
DMult	$c166	x,y-ZeroPage addr	uWORD(x) * uWORD(y) in uWORD(x)	*(x) produkt
Ddiv	$c169	x,y-ZeroPage addr	uWORD(x) DIV uWORD(y) in uWORD(x) r8	=*(x) MOD *(y)
			wWORD(x) MOD uWORD(y) in r8	*(x) =*(x) DIV *(y)
DSdiv	$c16c	x,y-ZeroPage addr	idem, maar met 2 signed WORDs	idem

notatie: sWORD(x) = signed word waarnaar de waarde van x wijst(in de ZeroPage,want x <= 255)
uBYTE(y) = unsigned byte waarnaar de waarde van y wijst.
*(x) = de waarde van het adres waarnaar x wijst

Illustratie 2

GEOS PROCES ROUTINES

naam routine	adres	variabelen in	werking	variabelen uit
InitProcesses		a-aantal, r0-tabel	init de processen, NIET gestart!	geen.
RestartProcess		x-procesnummer	start proces x (1e proc. => x=0)	geen.
BlockProcess		x-procesnummer	idem, maar blokkeer nu proces 1	geen.
UnblockProcess		x-procesnummer	idem, maar unblokkeer	geen.
FreezeProcess		x-procesnummer	idem, maar bevries	geen.
UnfreezeProcess		x-procesnummer	idem, maar ontdooi	geen.
Sleep		r0-slaaptijd in 50-ste seconden	volgt de laatste RTS terug maar keert als proces na slaaptijd naar aanroeper terug	geen.
EnableProcess		x-procesnummer	voer proces x eenmalig uit.	geen.

Illustratie 3

complementsysteem gaat door in (getal-1) alle bits te inverteren.

Andere systemen

Er zijn ook andere getalsystemen, waarvan ik er twee zal noemen: het 1-complementsysteem en het sign-magnitudesysteem. Deze systemen hebben hun specifieke voordelen, maar ook hun nadelen. Het 1-complementsysteem heeft als voordeel dat negatieveren erg makkelijk is: het inverse getal is gelijk aan het negatieve getal. Nadelen zijn, dat er twee nullen zijn: (000 is nul maar 111 ook), en dat optellen en aftrekken nu via een ander systeem gaat. Bijvoorbeeld: -1 + -1 = 110 + 110 =? 101 = -2. Het tweede alternatieve systeem, het sign-magnitudesysteem voegt 1 bit toe aan een getal dat aangeeft of het getal plus of min is. Op het eerste gezicht een helder systeem: het is zo te zien wat de waarde van een getal is. Bijvoorbeeld 101 is -1, want 1 (min) en 01. Nadelen zijn echter weer het onhandige optellen en aftrekken en het bestaan van twee nullen. Al met al is het 2-complementsysteem het meest gebruikte in computers.

Reken-routines

In illustratie 2 vindt U een tabel met GEOS reken-routines. Lang niet alle GEOS reken-operaties werken met negatieve getallen, dus U moet altijd goed kijken bij het 'werking'-veld. Alle operaties werken wat gegevens-invoer als volgt: de gebruiker moet x met ldx een waarde geven. Deze waarde is (natuurlijk) tussen 0 en 255. Dat is dus (als adres geinterpreteerd) een adres in de Zero Page. (De Zero Page is het geheugengebied van de C-64 bestaande uit de eerste 256 bytes, logisch dus). In dit adres moet U de gegevens dan zetten, d.w.z. als de invoer een byte is, zet dan Uw byte die U bijvoorbeeld wilt vermenigvuldigen in dat adres in de Zero Page. Als de invoer voor de routine een word is, dan zet U dat word dus in het adres in de ZeroPage en in het daaropvolgende adres. Vervolgens kan dan een routine uit de tabel aangeroepen worden.

Random

Een buitenbeentje in de tabel van illustratie 2 is de functie GetRandom. Deze functie trekt een willekeurig word en zet dit in de GEOS-variabele 'random'. Het adres van dit 'random' word is $850a/$850b. Volgens de GEOS Reference Guide heeft GetRandom een ongeveer uniform verdeeld uitkomstenpatroon (alle mogelijke 65536 uitkomsten komen gemiddeld even vaak uit). Gezien mijn ervaringen met het 'bouncing bars' programma lijkt dat inderdaad aardig te kloppen.

De routines BBMult, DMult en BMult voeren een vermenigvuldiging uit, de routine Dnegate negatieveert, Dabs geeft de absolute waarde (negatieve getallen worden positief, positieve blijven positief). De routine Ddec vermindert een word met 1, en Ddiv en DSdiv voeren een deling uit. Deze deling is echter geen deling met cijfers achter de komma: bij a en b en jsr Ddiv wordt het aantal malen dat b uit a kan teruggegeven (de wiskundige DIV functie). Het restant dat dan van a overblijft staat in r8 (de wiskundige MOD functie). Een voorbeeld: 17 Ddiv 5 is 3 (de DIV) met restant 2 (de MOD).

Macro's

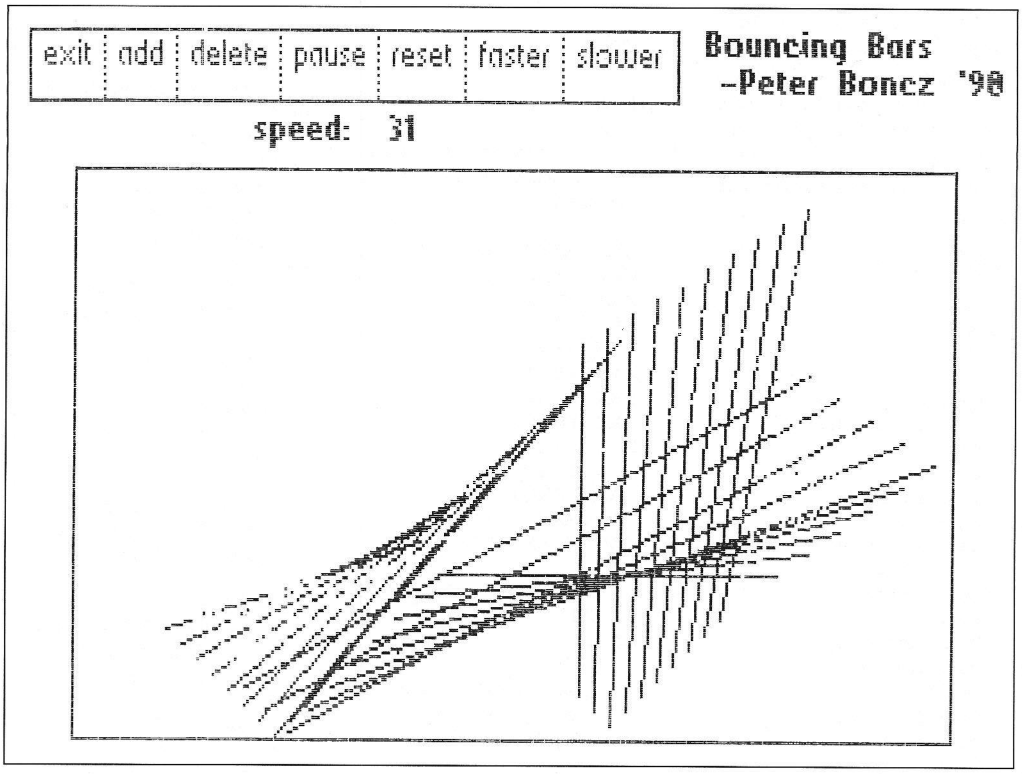
Nu denkt U misschien: maar waar zijn nu de optel- en aftrekroutines? Dat is heel simpel: ze staan in de geosMac file als macros. Mits U aan het begin van Uw programma een '.include geosMac' doet, kunt U gebruik maken van AddW w1,w2 en SubW w1,w2. AddW telt het eerste word (w1) bij het tweede 9w2) op en zet de uitkomst in w2, w1 blijft onveranderd. SubW trekt het eerste word (w1) van het tweede (w2) af en zet de uitkomst in w2, w1 blijft onveranderd. En omdat het twee-complementsysteem de optel- en aftrekinstrukties intact laat kunnen deze twee macro's voor zowel positieve als negatieve getallen gebruikt worden.

Problematisch

En dat is eigen alles wat over rekenen in GEOS gezegd kan worden. Hoe het allemaal in de praktijk gaat is in listing 1 te zien. Vooral in de routines 'rnd:' en 'tel_op:' wordt wat afgerekend. De eerstgenoemde routine trekt een willekeurig signed of een unsigned word (signed als de carry bit 1 is bij aanroep, unsigned als die 0 is) tussen een minimum en een maximum waarde. Dit gaat zo: eerst wordt GetRandom aangeroepen. Als er een signed word getrokken moest worden, dan wordt het hoogste bit onthouden in de variabele sign. Dit hoogste bit van random wordt standaard dan op nul gezet. Vervolgens wordt het random getal (inmiddels verplaatst in register a1) met (max-min) vermenigvuldigd. Dit gaat in 4 etappes van vermenigvuldiging met BBMult. Immers: bij een word maal word vermenigvuldiging krijg je een produkt van 4 bytes. Tenslotte wordt het getal door 32768 gedeeld en wordt er (min) bij opgeteld. De uitkomst is dan gegarandeerd tussen min en max. Als allerlaatste handeling wordt dan het getal genegativeerd als het in sign onthouden teken 1 is. Zoals U ziet blijft rekenen in machinetaal toch een problematische aangelegenheid. Met de GEOS functies erbij is het een heel stuk minder moeilijk, maar makkelijk wordt het nooit.

Processen

GEOS ondersteunt het werken met processen. Dat is leuk, denkt U nu misschien, maar wat zijn processen? Een proces is een routine die iedere N interrupts aangeroepen



Illustratie 4

wordt en een bepaalde functie uitvoert. N is hierin een door de programmeur ingestelde variabele. Zoals in vorige afleveringen aan de orde is gekomen bevindt 'de computer' zich als een applicatieprogramma draait doorgaans NIET in de applicatie, maar in de GEOS kernal: in de MainLoop. Dit is een lijst van check-handelingen, die als er iets gebeurt, tot actie overgaan, maar doorgaans niets doen en steeds maar weer achter elkaar door afgecheckt worden. Bijvoorbeeld: MainLoop kijkt of er een menu is geselecteerd of een icon is aangeklikt. Maar let op: MainLoop kijkt niet naar de hardware (in de twee genoemde voorbeeldjes is dat de muis of joystick) maar naar een eigen verborgen systeemvariabele om te bepalen of er een menu of een icon aangeklikt is. Deze systeemvariabelen worden gezet door InteruptLevel, dit is de welbekende $0314/$0315 interrupt van 50 keer in de seconde. InterruptLevel kijkt naar de hardware en zet overeenkomstig de systeemvariabelen, en de eerstvolgende keer dat MainLoop daar langs komt wordt er aktie ondernomen.

Werking

Processen worden gebruikt om een handeling 'continue' door te laten lopen, bijvoorbeeld het aanpassen van de tijd in een klok in de hoek van het scherm, terwijl er een applicatie loopt. De programmeur initialiseert en start dan een proces dat bijvoorbeeld iedere 50 interrups van InterruptLevel wordt aangeroepen (dus iedere seconde), en de secondenwijzer aanpast. Voor ieder proces dat geinitialiseert is houdt MainLoop een tellervariabele bij. Iedere interrupt wordt die teller 1 verlaagd. Als de MainLoop een 0 ziet, wordt de beginwaarde weer hersteld, en wordt de procesroutine aangeroepen. De procesroutine is een gewone machinetaalroutine (geschreven door U, in het voorbeeldgeval past hij de secondenwijzer aan). In illustratie 3 staan de GEOS routines om met processen te werken opgesomd. Om Uw processen te initialiseren moet er eenmaal InitProcesses aangeroepen worden, met in a het aantal processen en in r0 een zg. proces-tabel. Deze ziet er bijvoorbeeld zo uit:

```
.word Process1Routine
.word N1 ; dient elke N1 interrupts uitgevoerd te worden
.word Process2Routine
.word N2
etc. etc..
```

Als InitProcesses is aangeroepen lopen de processen nog niet. Ze kunnen gestart worden met de routine RestartProcess. In x moet het processnummer worden meegegeven. Let wel op: in alle proces- routines geeft x=0 het eerste proces aan, x=1 het tweede, enzovoorts.

Blokkeren

Processen kunnen geblokkeerd en bevroren worden. Het verschil is, dat bij blokkeren de teller iedere interrupt wel wordt aangepast, maar dat wanneer 0 bereikt wordt, er geen aktie ondernomen wordt. Bij een bevroren proces wordt de teller gewoon stopgezet. In het geval van een klok is het veel beter te blokkeren, omdat dan de 'tijd' blijft doorlopen. Andere soorten processen kunnen weer beter bevroren worden. Om een proces te blokkeren is er de routine BlockProcess. Herstarten van een geblokkeerd proces gaat met de routine UnblockProcess. Voor bevriezen/ontdooien zijn er FreezeProcess, en UnfreezeProcess. Bij al deze routines moet weer in x het procesnummer worden meegegeven.

Sleep

Tenslotte zijn er nog twee vreemde eenden in de bijt: Sleep en EnableProcess. De laatste is de simpelste: hij voert een lopend, geblokkeert of bevroren (maakt niet uit) proces eenmalig uit, en laat het lopend, geblokkeert of bevroren (wat het was). In x moet hier weer het procesnummer staan. Met de Sleepfunctie is het wat moeilijker. Deze routine doet wanneer hij aangeroepen wordt net alsof er een RTS staat. Na een slaaptijd van N interrupts, (N was in r0 voor bij de aanroep van Sleep meegegeven), keert GEOS echter bij de code direct achter de 'jsr Sleep' terug als eenmalig proces! Met Sleep moet U echter wel erg voorzichtig zijn, want U weet nooit vantevoren of de routine in de slaaptijd niet overschreven is. Ook alle variabelen x,y,a en ook alle registers zijn van waarde veranderd. Zorg er dus ten eerste voor dat het geheugen van de Sleep-routine niet overschreven wordt, en ten tweede dat de Sleep-routine totaal omgeving-onafhankelijk is.

Bouncing-bars

Het 'bouncing-bars' programma laat een of meer ta-

pijtjes van lijnen over het scherm dansen. Voor het uitrekenen van de coordinaten van de lijnen worden de GEOS-rekenroutines gebruikt. Voor het aanpassen, tekenen en wissen van de lijnen wordt een proces gebruikt. In listing 1 vlak voor de menu-tabel staat de 'processTable:', met eerst een pointer naar de proces-routine ('handleBars'), en vervolgens de frequentie waarmee deze aangeroepen moet worden door GEOS (in het begin zoals U ziet iedere 36 interrupts). Het verdere programma bestaat daarnaast, geheel onafhankelijk, uit een horizontaal menu met zeven opties: 'exit', 'add' (1 tapijt meer op het scherm), 'delete' (1 minder), 'pause' (om het beeld stil te zetten), 'reset' (trekt nieuwe coordinaten voor alle lopende tapijten), en 'faster' (laat de tapijten sneller lopen), en tenslotte 'slower'. De functie 'pause' voert dus een BlockProcess uit, of een UnblockProcess als het beeld al stil stond.

Compileren

Alle listings dienen in GeoWrite ingetikt te worden: listing 1 onder de naam 'barLst', en listing 2 dient 'barLnk' te heten. Listing 1 moet met de GeoAssembler gecompileerd worden. Listing 2 is bestemd voor de GeoLinker. In listing 3 tenslotte staat een macro dat in het 'bouncing-bars' programma gebruikt wordt. Het is de bedoeling dat U dit macro aan het begin van de 'geosMac' file zet. Het macro heet 'jsm', en heeft als parameters 'wrd' (een adres van een word), 'value' (een getal tussen -32768 en 32767) en 'addr' (een adres om naartoe te springen). De betekenis van jsm is JumpSMaller: jsm wrd,value,addr springt naar 'addr' als de waarde van 'wrd' kleiner is dan het getal 'value'. Het macro wordt in listing 1 gebruikt om te kijken of de coordinaten van een bewegend tapijt niet buiten het beeld dreigen te gaan vallen. Dit gebeurt onderaan in de routine 'tel_op:'. En nu rest mij niets anders U nog veel plezier met het programma te wensen en tot een volgende aflevering afscheid te nemen.

Peter Boncz

LIST 3

```
.macro jsm
  wrd,value,addr
 php
 pha
 lda wrd+1
 cmp #0
.if (value=0) &&
   (value)
 bmi jump
.else
 beq nothing
 bpl nothing
.endif
 cmp #](value)
 beq cont
 bcc jump
 bcs nothing
cont:
 lda wrd
 cmp #[(value)
 bcs nothing
jump:
 pla
 plp
 jmp addr
nothing:
 pla
 plp
```

list 2

```
.output bounce
.psect $0800
.ramsect $1700
barLst.rel
```

LIST 1

```
pl 127
 .if Pass1
     .include geosSym
     .include geosMac
 .endif

 MAX_BARS=16
 MAX_LINES=10
 MAX_VX=16
 MAX_VY=10
 MIN_VX=1
 MIN_VY=1
 MAX_X=300
 MAX_Y=190
 MIN_X=20
 MIN_Y=40
 BAR_LENGTH=
  10+MAX_LINES*8
 TABLE_LENGTH=

    MAX_BARS*BAR_LENG
    TH
 main:
    lda #0
    jsr SetPattern
    jsr i_Rectangle
    .byte 0,199
    .word 0,319
    LoadW r0,menu
    jsr DoMenu
    LoadW r1,
         barsTable
    LoadW r0,
         TABLE_LENGTH
    jsr ClearRam
    jsr i_PutString
    .word 220
    .byte
   8,24,"Boun"
    .byte "cing
   Bars"
    .byte 10,8,8,8,8
    .byte 8,"-P. Bo"
    .byte "ncz
   '90",0
    jsr

    i_FrameRectangle
    .byte MIN_Y-1,
    .byte MAX_Y+1
    .word MIN_X-1
    .word MAX_X+1
    .byte 255
exec_bars:
    jsr write_speed
    LoadW r0,

    procTable
    lda #1
    jsr
    InitProcesses
    ldx #0
    jsr

    RestartProcess
    jmp MainLoop
exit:
    ldx #0
    jsr
    FreezeProcess
    jmp EnterDeskTop
add:
    jsr add_bar
roll_menu:
    jsr ReDoMenu
    jsr MouseUp
    jmp MainLoop
delete:
    jsr delete_bar
    jmp roll_menu
pause:
    lda blocked
    cmp #0
    beq 1$
    ldx #0
    jsr

    RestartProcesses
    LoadB blocked,0
    jmp roll_menu
1$:
    ldx #0
    jsr BlockProcess
    LoadB blocked,1
    jmp roll_menu
reset:
    LoadB a7H,0
1$:
    jsr delete_bar
    cpy #0
    beq 2$
    inc a7H
    bra 1$
2$:
    lda a7H
    cmp #0
    beq 3$
    dec a7H
    jsr add_bar
    bra 2$
3$:
    lda #0
    jsr SetPattern
    jsr i_Rectangle
    .byte
   MIN_Y,MAX_Y
    .word
   MIN_X,MAX_X
    jmp roll_menu
slower:
    clc
    lda #8
    adc procTable+2
    sta procTable+2
    cmp #252
    bne 1$
    lda #244
    sta procTable+2
1$:
    jmp exec_bars
faster:
    lda procTable+2
    cmp #4
    beq 1$
    sec
    sbc #8
    sta procTable+2
1$:
    jmp exec_bars
```

```
write_speed:
    lda #0
    jsr i_Rectangle
    .byte 23,37
    .word 70,200
    LoadW r11,75
    LoadB r1H,31
    LoadW
    r0,speedTxt
    jsr PutString
    LoadW r0,260
    SubW procTable+2
                 ,r0
    lsr r0H
    ror r0L
    lsr r0H
    ror r0L
    lsr r0H
    ror r0L
    lda #192
    jsr PutDecimal
    jsr ReDoMenu
    jsr MouseUp
    rts
handleBars:
    LoadW a0,

   barsTable
    ldx #MAX_BARS
0$:
    jsr get_bar
    cpy #0
    beq 1$
    txa
    pha
    jsr draw_bar
    pla
    tax
    bra 0$
1$:
    rts
max:
    .word 0
min:
    .word 0
sign:
    .byte 0
rnd:
    lda #0
    sta sign
    adc sign
    sta sign
    SubW min,max
0$:
    jsr GetRandom
    MoveW random,a1
    lda sign
    cmp #0
    beq 1$
    asl a1H
    lda #0
    sta sign
    adc sign
    sta sign
    bra 2$
1$:
    asl a1H
2$:
    lsr a1H
    asl max
    rol max+1
    ldy #a1H
    ldx #a2L
    MoveB max,a2L
    jsr BBMult
    ldy #a1L
    ldx #a3L
    MoveB max+1,a3L
    jsr BBMult
    clc
    lda a2L
    adc a3L
    sta a3L
    lda a2H
    adc a3H
    sta a3H
    lda #0
    sta r4L
    sta r4H
    adc r4H
    sta r4H
    ldy #a1L
    ldx #a2L
    MoveB max,a2L
    jsr BBMult
    lda a2H
    clc
    adc a3L
    lda #0
    adc a3H
    sta a4L
    lda #0
    adc r4H
    sta r4H
    ldy #a1H
    ldx #a2L
    MoveB max+1,a2L
    jsr BBMult
    AddW a4,a2
    AddW min,a2
    lda sign
    cmp #0
    beq 3$
    ldx #a2
    jsr Dnegate
3$:
    ldy #0
    lda a2L
    sta (a0),y
    lda a2H
    iny
    sta (a0),y
inc2_a0:
    clc
    inc a0L
    inc a0L
    lda #0
    adc a0H
    sta a0H
    rts
wissen:
    ldy #0
    lda (a4),y
    sta r3L
    iny
    lda (a4),y
    sta r3H
    iny
    lda (a4),y
    cmp #0
    beq 1$
    sta r11L
    iny
    iny
    lda (a4),y
    sta r4L
    iny
    lda (a4),y
    sta r4H
    iny
    lda (a4),y
    lsr r11H
    sta r11H
    clc
    jsr DrawLine
1$:
    rts
draw_bar:
    ldy #1
    lda (a6),y
    sta a5L
    lda #0
    sta a5H
    asl a5H
    rol a5H
    asl a5H
    rol a5H
    asl a5H
    rol a5H
    lda #2
    adc a5L
    sta a5L
    lda #0
    adc a5H
    sta a5H
    AddW a6,a5
    ldy #1
    lda (a6),y
    cmp #MAX_LINES
    beq 1$
    MoveW a5,a4
    lda #8
    bra 2$
1$:
    MoveW a6,a4
    lda #10
2$:
    clc
    adc a4L
    sta a4L
    lda #0
    adc a4H
    sta a4H
    jsr wissen
    LoadB a7H,2
tel_op:
    ldy a7H
    lda (a6),y
    sta a8L
    iny
    lda (a6),y
    sta a8H
    dey
    dey
    lda (a5),y
    sta a9H
    dey
    lda (a5),y
    sta a9L
    AddW a8,a9
    ldx #a8
    jsr Dnegate
    lda a7H
    cmp #2
    bne 1$
0$:
    jmp l1
1$:
    cmp #6
    beq 0$
    jsm a9,MIN_Y,
          omdraaien
    jsm a9,MAX_Y+1,
          store
    jmp omdraaien
l1:
    jsm a9,MIN_X,
          omdraaien
    jsm a9,MAX_X+1,
          store
omdraaien:
    ldy a7H
    lda a8L
    sta (a6),y
    iny
    lda a8H
    sta (a6),y
    jmp tel_op
store:
    ldy a7H
    dey
    lda a9H
    sta (a4),y
    dey
    lda a9L
    sta (a4),y
    ldy a7H
    iny
    iny
    sty a7H
    cpy #10
    beq teken
    jmp tel_op
teken:
    ldy #1
    lda (a6),y
    tax
    inx
    cpx #MAX_LINES+1
    bne 1$
    ldx #1
1$:
    txa
    sta (a6),y
    ldy #0
    lda (a4),y
    sta r3L
    iny
    lda (a4),y
    sta r3H
    iny
    lda (a4),y
    sta r11L
    iny
    iny
    lda (a4),y
    sta r4L
    iny
    lda (a4),y
    sta r4H
```

```
    iny
    lda (a4),y
    lsr r11H
    sta r11H
    sec
    jsr DrawLine
    rts
get_bar:
    ldy #0
    cpx #0
    beq 1$
    MoveW a0,a6
    clc
    lda #[BAR_LENGTH
    adc a0L
    sta a0L
    lda #]BAR_LENGTH
    adc a0H
    sta a0H
    dex
    lda (a6),y
    cmp #1
    bne get_bar
    ldy #1
1$:
    rts
delete_bar:
    ldx #MAX_BARS
    LoadW a0,

   barsTable
    jsr get_bar
    cpy #0
    beq uit
    lda #MAX_LINES
    pha
    MoveW a6,a4
    lda #10
1$:
    clc
    adc a4L
    sta a4L
    lda #0
    adc a4H
    sta a4H
    jsr wissen
    pla
    tax
    dex
    beq 2$
    txa
    pha
    lda #8
    bra 1$
2$:
    LoadW r0,
          BAR_LENGTH
    MoveW a6,r1
    jsr ClearRam
uit:
    rts
add_bar:
    ldy #0
    ldx #0
    LoadW a0,

   barsTable
1$:
    lda (a0),y
    cmp #0
    beq 3$
    inx
    cpx #MAX_BARS
    beq uit
    clc
    lda
    #[BAR_LENGTH

    adc a0L
    sta a0L
    lda #]BAR_LENGTH
    adc a0H
    sta a0H
    bra 1$
3$:
    lda #1
    sta (a0),y
    iny
    sta (a0),y
    jsr inc2_a0
    MoveW a0,r1
    LoadW r0,
        BAR_LENGTH-2
    jsr ClearRam
    LoadB a6L,2
4$:
    LoadW min,MIN_VX
    LoadW max,MAX_VX
    sec
    jsr rnd
    LoadW min,MIN_VY
    LoadW max,MAX_VY
    sec
    jsr rnd
    clc
    dec a6L
    bne 4$
    LoadB a6L,2
5$:
    LoadW min,MIN_X
    LoadW max,MAX_X
    clc
    jsr rnd
    LoadW min,MIN_Y
    LoadW max,MAX_Y
    clc
    jsr rnd
    dec a6L
    bne 5$
    rts

procTable:
    .word handleBars
    .word #36
blocked:
    .byte #0
menu:
    .byte 2,21
    .word 4,212
    .byte 7
    .word exitTxt
    .byte 0
    .word exit
    .word addTxt
    .byte 0
    .word add
    .word deleteTxt
    .byte 0
    .word delete
    .word pauseTxt
    .byte 0
    .word pause
    .word resetTxt
    .byte 0
    .word reset
    .word fasterTxt
    .byte 0
    .word faster
    .word slowerTxt
    .byte 0
    .word slower
exitTxt:
    .byte "exit",0
addTxt:
    .byte "add",0
deleteTxt:
    .byte "delete",0
pauseTxt:
    .byte "pause",0
resetTxt:
    .byte "reset",0
fasterTxt:
    .byte "faster",0
slowerTxt:
    .byte "slower",0
speedTxt:
    .byte
   "speed:",0
.ramsect
 barsTable:
 .block
   TABLE_LENGTH
```

AMIGA
C Gadgets

Het weer heeft deze zomer niet echt bijgedragen aan het opzetten van baanbrekende softwarecreaties. Temperaturen boven de 30 graden creëren immers niet meteen 'het lekker binnenhuis met de computer beziggaan'-gevoel. Daarom gooien we nu de trossen los om het Amiga-ritme weer te pakken te krijgen.

In de voorgaande aflevering hebben we geprobeerd u enig inzicht te geven in Intuition-functies ten aanzien van tekst en graphics. Aansluitend hierop gaan we nu een stap verder. We gaan ons bezighouden met Gadgets welke door Intuition worden beheerd.

Wat zijn Gadgets

Gadgets zijn symbolen op het beeldscherm welke, als ze geactiveerd worden door de gebruiker, een signaal afgeven aan Intuition. Het activeren van een Gadget gebeurt door middel van het aanwijzen van het Gadget met de muis waarna de linker muisknop ingedrukt wordt.
Een Gadget kan bestaan uit alleen een kaal rechthoekje, of een rechthoekje voorzien van tekst. Hier blijft het niet bij, het is ook mogelijk zelf gedefinieerde grafische symbolen te laten verschijnen binnen de Gadget ruimte.
Welke soorten Gadgets kunnen we onderscheiden?
1. Boolean gadgets.
2. Proportional gadgets.
3. String gadgets.

```
Figuur 1.
De Gadget-structuur

struct Gadget {
 struct Gadget
    *NextGadget;
 SHORT LeftEdge;
 SHORT TopEdge;
 SHORT Width;
 SHORT Height;
 USHORT Flags;
 USHORT Activation;
 USHORT GadgetType;
 APTR GadgetRender;
 APTR SelectRender;
 struct IntuiText
    *GadgetText;
 LONG
    MutualExclude;
 APTR SpecialInfo;
 USHORT GadgetID;
 APTR UserData;
};
```

Figuur 1

Booleaanse of logische Gadgets geven alleen ja/nee antwoorden terug. Ingedrukt, wordt er een signaal verzonden dat het gadget ingedrukt is. Met dit type Gadget kunt u dus simpele ja/nee of aan/uit Gadgets programmeren.
Proportionele Gadgets dienen als tegenhanger van de in analoge technieken bekende schuifregelaar. De proportionele Gadget kan om deze reden gebruikt worden als men een meer 'intuitieve' invoermogelijkheid.
StringGadgets tot slot, geven de gebruiker de mogelijkheid om tekst in te voeren. De stringgadgets kan bekeken worden als de ultieme invoerroutine. Intuition zorgt ervoor dat er tijdens de invoer niets misgaat. Het enigste dat de gebruiker dient te doen is het Gadget aanklik-

```
Figuur 2. De bij het element Flags
    behorende bits.

De bits zijn onder te verdelen in een
    aantal groepen. Die bits die bepalen
    hoe het Gadget eruit ziet in
    geactiveerde toestand, gegevens omtrent
    de positie van het Gadget en de bits
    die de status van het Gadget
    definieëren.

Groep 1GADGHCOMPBij aanklikken wordt het
    Gadget vertoont in de complementaire
    kleuren.
GADGHBOXBij aanklikken verschijnt er een
    rechthoek om het Gadget.
GADGHIMAGEDe gebruiker wil een zelf
    gedefinieerde Image afbeelden op de
    plaats van het Gadget.
GADGHNONEBij aanklikken dient er niets te
    gebeuren op het beeldscherm.
Groep 2GRELBOTTOMAls deze flag gebruikt
    wordt zal het Gadget gepositioneerd
    worden tov. de onderkant van het
    scherm, als ze niet gezet wordt wordt
    het Gadget gepositioneerd tov. de
    bovenkant van het scherm.
GRELRIGHTBij gebruikte flag zal het Gadget
    gepositioneerd worden tov. de
    rechterkant van het scherm, als de flag
    niet gebruikt wordt zal het Gadget
    gepositioneerd tov. de linkerkant van
    het scherm.
GRELWIDTHDe grootte van het Gadget wordt
    bepaalt mbv. de breedte van het scherm.
    De breedte van het Gadget = breedte van
    het Window - waarde van het element
    Width in de Gadget-structuur.
GRELHEIGHTDe hoogte van het Gadget wordt
    bepaalt mbv. de hoogte van het scherm.
    Gadgethoogte = hoogte van het Window -
    waarde van het element Height in de
    Gadget-structuur.
Groep 3GADGIMAGEHierbij zal er een
    zelfgedefinieerd Image afgebeeld worden.
GADGDISABLEDMet behulp van dit flag zet u
    een Gadget buiten werking. Deze flag
    heeft alleen effect voordat u het
    Gadget activeert.
SELECTEDAls deze flag gezet is, is het
    Gadget geselecteerd.
```

Figuur 2

ken en het invoeren van tekst kan beginnen.

Een Gadget programmeren

Alvorens men kan beginnen met het programmeren van een Gadget dient men enkele zaken te weten. We beginnen met de Gadget-structuur. **Figuur 1** bevat de gehele Gadget-structuur. Met behulp van deze structuur kan men het af te beelden Gadget beinvloeden. Het eerste element van de Gadget-structuur slaan we in eerste instantie over, dit komt later in deze aflevering aan de orde. De volgende vier elementen dienen ondertussen al bekend te zijn, ze omvatten de linkertophoek coördinaten, de breedte en de hoogte. De elementen Flags, Activation en GadgetType worden uitgelegd in resp. **figuur 2 en 3**. Het gaat hier nl. om een serie flags die apart worden behandelt. GadgetRender dient een pointer naar een Imagestructuur te bevatten indien u een door u zelf gedefinieerd plaatje af wilt beelden. Wordt geen Image gewenst dan zal hier een pointer naaar een Border structuur moeten staan. SelectRender. Indien het Gadget er in 'aangeklikte' toestand verschillend van de originele toestand uit dient te zien, dan moet hier de pointer naar de betreffende Image-structuur vermeld staan. In het elfde element, GadgetText, moet een pointer naar een IntuiTextstructuur staan als u tekst in of buiten het Gadget afgebeeld wilt zien. In de IntuiText structuur definieert u vervolgens de eigenschappen van de tekst, zoals positie en kleur. Het element MutualExclude heeft in de huidige WorkBench nog geen functie. Ze is opgenomen vanwege compatibiliteit met toekomstige versies. SpecialInfo moet een pointer bevatten als men gebruik wil maken van Proporionele- of Stringgadgets. Bij een Proportioneel Gadget dient SpecialInfo een pointer naar een PropInfo-structuur te bevatten, bij een String Gadget moet het element wijzen naar een StringInfo-structuur. Op beide zaken komen we straks uitgebreid terug. In het element GadgetID kunt u een door u zelf aan te geven nummer invullen. Dit nummer zal vervolgens altijd terug komen als u de Gadgets via Intuition uitleest. Het element UserData is binnen het bestek van deze cursus niet van belang en zal daarom ook niet nader uitgelegd worden.

Logische Gadgets

Voordat we verder gaan met het programmeren van een logisch Gadget, eerst enige uitleg omtrent het activeren van een Gadget. Een Gadget kan op twee manieren gestart worden. Ten eerste kan dit door in de NewWindow-structuur de pointer naar het eerste Gadget te vermelden. Het element FirstGadget in de NewWindow-structuur dient deze pointer te bevatten. Wilt u meerdere Gadgets starten dan zult u de Gadgets aan elkaar moeten 'koppelen'. Dit kan door gebruik te maken van het element NextGadget in de Gadget-structuur. Hier dient de pointer naar het volgende Gadget te staan. Is de rij compleet dan dient u de NextGadget pointer van het laatste Gadget op NULL te zetten. De tweede manier. Start één Gadget op met de functies AddGadget() en RefreshGadgets() of, in het geval van meerdere Gadgets, start de gekoppelde Gadget-lijst op met AddGList() en RefreshGList(). Deze vier functies komen bij de bespreking van de Gadgetfuncties straks nog aan de orde.

Een logisch of booleaans Gadget kunt u programmeren door in het element GadgetType, BOOLGADGET in te vullen. Vervolgens kunt u visuele eigenschappen, zoals Images, Borders en Teksten, van het Gadget programmeren. Images worden vertoont als de flag GADGHIMAGE in het element flags gezet is, anders zal het Gadget een Border-structuur verwachten. Tekst bij een Gadget zal vertoont worden als er een pointer naar een IntuiText-structuur gezet is in het element GadgetText. Tot slot, de wijze waarop het Gadget reageert op het aanklikken met de muis, wordt bepaalt door een aantal flags. Deze

```
Figuur 3. De flags van het element
    Activation.

TOGGLESELECTHet Gadget wordt een
    wisselschakelaar. De eerste keer dat er
    gedrukt is, zal het Gadget op AAN
    staan, bij de volgende druk zal het
    Gadget op UIT staan.
GADGIMMEDIATEBij het aanklikken van het
    Gadget krijgt het programma dat het
    Gadget geactiveerd heeft, meteen een
    signaal.
RELVERIFYHet programma krijgt pas een
    signaal wanneer de gebruiker het Gadget
    losgelaten heeft.
RIGHTBORDERHet Gadget komt in de
    rechterrand van het
    GimmeZeroZerovenster te staan.
LEFTBORDERHet Gadget komt in de linkerrand
    van het GimmeZeroZeroVenster te staan.
TOPBORDERHet Gadget komt in de bovenrand
    van het venster te staan.
BOTTOMBORDERHet Gadget komt in de
    onderrand van het venster te staan.
STRINGCENTERBij stringgadgets worden de
    strings in het midden van het
    invoervenster geplaatst. Dit gebeurt
    continu, dus ook tijdens de invoer.
STRINGRIGHTBij stringgadgets, hier worden
    de strings aan de rechterkant van het
    invoervenster afgebeeld.
LONGINTBij stringgadgets, in plaats van
    strings wordt nu een integerwaarde
    verwacht in het invoervenster.
ALTKEYMAPBij stringgadgets, er wordt een
    eigen toetstenbordtabel gebruikt. Vul
    de pointer naar een andere
    toetsenbordconfiguratie in, in het
    element AltKeyMap van de structure
    StringInfo.
ENDGADGETBij gebruik van requesters. Als
    een Gadget met gezet ENDGADGET-bit
    ingedrukt is dan zal het requester in
```

Figuur 3



flags worden nader behandelt in **figuur 2**.

Proportionele Gadgets

De logische- of booleaanse Gadgets zijn het eenvoudigst te programmeren. Bij proportionele Gadgets komt er iets meer om de hoek kijken. Ook bij bij proportionele Gadgets dient men van een Gadget-structuur gebruik te maken. Naast deze structuur dienen we nog van een andere structuur gebruik te maken. De PropInfo-structuur, zie **figuur 5,** bevat een aantal extra gegevens omtrent de schuifcontainer, het vierkante blokje binnen een proportionele Gadget, zoals grootte en positie. Het eerste element van de PropInfo-structuur, Flags genaamd, kan ingevuld worden met één of meerdere van de volgende vijf flags.
AUTOKNOBHierdoor wordt de standaard schuifcontainer, geleverd door Intuition, geaccepteerd. Alle afmetingen van de schuif worden hierdoor automatisch aangepast.
FREEHORIZDe schuifcontainer kan horizontaal verplaatst worden.
FREEVERTDe schuifcontainer kan verticaal verplaatst worden. (Als zowel FREEHORIZ als FREEVERT gezet is zal er een tweedimensionaal proportioneel Gadget ontstaan.)
KNOBHITDeze flag wordt door Intuition beheerd. U hoeft deze niet te gebruiken.
PROPBORDERLESS
De rand om de proportionele Gadget wordt niet getekend.

HorizPot en VertPot bevatten resp. de horizontale als vertikale postitie van de schuifcontainer. HorizBody en VertBody dienen resp. de horizontale en vertikale stapgrootte van de schuifcontainer te bevatten. Het volgende paar, CWidth en CHeight dienen zoals de naam al doet vermoeden, de breedte en de hoogte van de schuifcontainer te bevatten. HPotRes en VPotRes. In deze twee elementen van de structuur dienen de minimale stapgrootte van de schuifcontainer te staan. De laatste twee elementen van de PropInfo-structuur, LeftBorder en TopBorder, moeten de coördinaten van resp. de linker- en bovenrand van het schuifgedeelte van de proportionele Gadget bevatten.
In het element SpecialInfo van de Gadget-structuur vult u de pointer naar de PropInfo-structuur in. Hiermee kan Intuition de locatie van deze structuur in het geheugen vinden. De proportionele gadget is nu bijna geprogrammeerd. Er ontbreekt nog één ding, Intuition moet er nog op worden geattendeerd dat er een Proportioneel Gadget getekend moet worden. Dit bereikt u door het volgende te doen. Vul in het element GadgetType, PROPGADGET in. Intuition is hiermee op de hoogte van uw bedoelingen. Helaas, helaas, hiermee komt er nog geen Gadget op het scherm. Daar de proportionele gadget gebruik maakt van een Image, immers de schuifcontainer is een grafische afbeelding, een Image, moet in het element GadgetRender een pointer naar een LEGE Image-structuur staan. U hoeft niets meer te doen dan de Image-structuur te declareren. Intuition neemt de rest voor u uit handen.
Nog een opmerking wanneer u gebruik maakt van eigen afbeeldingen. Denk erom dat de Image data in Chip-geheugen staan, anders zal er niets op het beeldscherm verschijnen.

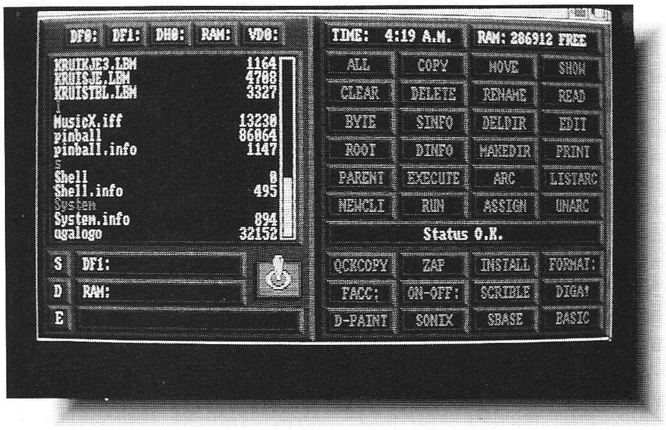
```
Figuur 4. De flags
    van het
    element
    GadgetType.

BOOLGADGETHet
    Gadget is een
    Booleaans
    Gadget.
PROPGADGETHet
    Gadget is een
    Proportioneel
    Gadget.
```

Figuur 4

String Gadgets

Ook bij string Gadgets geldt hetzelfde als bij de hiervoor besproken proportionele Gadgets. De Gadget-structuur dient ingevuld te worden, waarbij het element GadgetType op STRGADGET gezet dient te worden. In het Activation-veld kunt u nog enige andere stringgadget-eigenschappen vermelden zoals, STRINGCENTER of STRINGRIGHT. Hiermee zal de cursor in het invoervenster resp. in het midden of aan de rechterkant staan.
Ook bij de string Gadgets heeft Intuition niet genoeg aan de Gadget-structuur. De StringInfo-structuur bevat nog enige additionele elementen die ingevuld dienen te worden. Geef in het element SpecialInfo de pointer naar de StringInfo-structuur, vul de StringInfo-structuur in en uw Gadget is klaar om gestart te worden. Hoe ziet de StringInfo-structuur eruit? Bekijk **figuur 6** eerst eens. **Figuur 6** bevat de StringInfo-structuur zoals deze ook te vinden is in het bestand 'intuition/intuition.h' bij uw Lattice of Aztek C compiler. Het element Buffer moet een pointer bevatten naar een buffer waar de ingevoerde tekst in kan worden opgeslagen. Undobuffer dient, net als het element Buffer, een pointer naar een buffer te bevatten waarin de ongecorrigeerde tekst in kan worden opgeslagen. U hoeft dit element niet in te vullen, het is echter wel een extra service naar de gebruiker toe als u gebruik maakt van een UndoBuffer. Met behulp van het element BufferPos zorgt u ervoor dat de cursor op een bepaalde plaats in het invoervenster wordt geplaatst. MaxChars zal het maximale aantal karakters moeten bevatten. Gaat dit aantal boven het aantal karakters uit dat binnen het invoervenster past dan zal men binnen dit venster kunnen schuiven met de tekst net zolang tot u een MaxChars aantal karakters in het venster ingetypt hebt. DispPos, tot slot, dient de positie van de cursor in het displayveld te bevatten. De StringInfo-structuur bevat nog een aantal andere elementen. Deze elementen




```
Figuur 5. De
    PropInfo-structu
    ur.

struct PropInfo {
 USHORT Flags;
 USHORT HorizPot;
 USHORT VertPot;
 USHORT HorizBody;
 USHORT VertBody;
 USHORT CWidth;
 USHORT CHeight;
 USHORT HPotRes;
 USHORT VPotRes;
 USHORT LeftBorder;
 USHORT TopBorder;
```

Figuur 5

hoeft u niet in te vullen, Intuition beheert ze zelf. U kunt deze wel uitlezen. U kunt bijvoorbeeld in het geval van een LONGINT-stringgadget de waarde van het ingevoerde getal uit het element LongInt lezen.
Gewapend met deze kennis dienen we nu in staat te zijn een programma te maken, welke een aantal Gadgets beheerd.

Gadgets uitlezen

Voordat we het een en ander in praktijk brengen, willen we eerst nog wat zeggen over het uitlezen van Gadgets. Onder het uitlezen van Gadgets bedoelen we het verkrijgen van gegevens over de 'aangeraakte' Gadgets. We willen immers weten welk Gadget aangeraakt werd, wat voor tekst ingevoerd is of op welke positie de proportionele gadget terecht gekomen is.
Het leesprincipe berust op het uitlezen van de IDCMP-ports, de Intuition Direct Communications Message Ports. Met deze gegevens kan men vervolgens bepalen welk Gadget ingedrukt werd, waarna men de daarbijbehorende gegevens uit kan lezen. Het geheel draait om de functie GetMsg(). Deze functie wordt later in deze cursus behandeld, ga er voor nu maar van uit dat het dient om boodschappen van andere devices uit te lezen. De gegevens worden ingelezen in een structure van het type IntuiMessage. Hoe ziet deze structure eruit? (fig 7) De elementen van belang zijn,
Class geeft de IDCMP flag weer. Bij een Gadget-Message kan dit GADGETUP of GADGETDOWN zijn.
Code bevat een code geassocieerd met de boodschap, zoals een toetsencode.
Qualifier is bij toetsencodes van belang. Ze geeft de SHIFT/CTRL/ALT toetsencodes weer.
IAddress wijst naar een structuur die bij de boodschap hoort. Bij Gadgets hoort hier een pointer naar Gadget-

```
Figuur 6. De
    StringInfo-struct
    uur.

struct StringInfo {
 UBYTE *Buffer;
 UBYTE *UndoBuffer;
 SHORT BufferPos;
 SHORT MaxChars;
 SHORT DispPos;
 SHORT UndoPos;
 SHORT NumChars;
 SHORT DispCount;
 SHORT CLeft;
 SHORT CTop;
 struct Layer
    *LayerPtr;
 LONG LongInt;
 struct KeyMap
    *AltKeyMap;
};
```

Figuur 6

structuur te staan. De overige elementen worden in een volgend deel van de cursus nader bekeken.
Om een melding te ontvangen dient er te worden gewacht op de melding. Dit kan op de volgende wijze,

```
while(!(Boodschap=(str
    uct IntuiMessage
    *)GetMsg(Window->U
    serPort)));
```

waarbij:
struct IntuiMessage *Boodschap;
struct Window *Window;
Dit werkt uitstekend, echter het zorgt er wel voor dat in een multitasking omgeving zoals die van de Amiga, onevenredig veel tijd wordt weggestolen van de processor terwijl het proces niets anders staat te doen dan te wachten. Er moet dus een betere oplossing zijn.
Het antwoord komt vanuit de ExecBase library. Deze library levert de functie Wait() waarmee men een proces tijdelijk tot stilstand kan brengen totdat er een signaal optreed.

```
Wait(1L <<
    Window->UserPort->
    mp_SigBit);
Boodschap=(struct
    IntuiMessage *)

    GetMsg(Window->Use
    rPort);
ReplyMsg(Boodschap);
```

In het bovenstaande voorbeeld wordt een nieuw element geintroduceerd, namelijk de functie ReplyMsg(). U dient ALTIJD na een Message ontvangen te hebben, deze te replieen, omdat anders het versturende hopeloos vast blijft zitten, wat onvermijdelijk in een system shutdown zal resulteren. Denk er wel om dat er gecontroleerd moet worden of de Message ook werkelijk aanwezig is, dwz. de Message mag NIET gelijk aan nul zijn. Hiermee komen we tot de laatste verandering. De uiteindelijke uitleesroutine zou er als volgt uit kunnen zien,

```
Wait(1L <<
    Window->UserPort->
    mp_SigBit);

while(!(Boodschap=(str
    uct IntuiMessage
    *)

    GetMsg(Window->Use
    rPort)));
ReplyMsg(Boodschap);
```

Wait() zal in een volgend deel van de cursus nader worden uitgelegd, voor nu gaan we er van uit dat het proces wat deze functie hanteert, vast blijft zitten totdat er een specifieke actie voorkomt waarop Exec het proces door laat gaan.

```
Figuur 7. De
    IntuiMessage-stru
    ctuur.

struct
    IntuiMessage {
 struct Message
    ExecMessage;
 ULONG Class;
 USHORT Code;
 USHORT Qualifier;
 APTR IAddress;
 SHORT MouseX;
 SHORT MouseY;
 ULONG Seconds;
 ULONG Micros;
 struct Window
    *IDCMPWindow;
 struct
    IntuiMessage
    *SpecialLink;
};
```

Figuur 7

Gadget functies

Om de Gadgets te beheersen en te sturen dient men over meerdere functies te beschikken om dit uit te voeren. We behandelen de functies in alfabetische volgorde.

ActivateGadget()

De functie activeert een string Gadget.

```
Resultaat=ActivateGadg
    et(Gadget,Window,R
    equester);
BOOL Resultaat;
struct Gadget
    *Gadget;
struct Window
    *Window;
struct Requester
    *Requester;
```

Als de Gadget niet in een Requester aanwezig is dan kan de laatste parameter op NULL gezet worden.

```
Error=ActivateGadget(&
    MijnGadget,MijnWin
    dow,NULL);
```

waarbij:

```
BOOL Error;
struct Gadget
    MijnGadget;
struct Window
    *MijnWindow;
```

AddGadget()

Met deze functie wordt een Gadget toegevoegd aan de Gadgetlijst. Gewoonlijk wordt de parameter Position op -1 (min één) gezet. Dit betekent dat de Gadget achter aan de lijst wordt toegevoegd. Men kan ook een positie opgegeven zoals 0, 1, 2, e.a.

```
TBPositie=AddGadget(Wi
    ndow,Gadget,Positi
    e);
USHORT
    TBPositie,Positie;

struct Window
    *Window;
struct Gadget
    *Gadget;
Pos=AddGadget(MijnWind
    ow,&MijnGadget,-1)
    ;
```

AddGList()

Wanneer u meerdere Gadgets toe wilt voegen dient u gebruik te maken van de functie AddGList(). Voorwaarde is wel dat de lijst met Gadgets onderling met elkaar verbonden is.

```
TBPositie=AddGList(Win
    dow,Gadget,Positie
    ,AantalGadg,Reques
    ter);
USHORT
    TBPositie,Positie,
    AantalGadg;
struct Window
    *Window;
struct Gadget
    *Gadget;
struct Requester
    *Requester;
```

Ook hierbij geld weer, als er Gadgets binnen een Requester afgebeeld dienen te worden dan moet de pointer naar het Requester meegegeven worden.
In AantalGadg dient men het aantal Gadgets op te geven dat in de lijst opgenomen moeten worden.

ModifyProp()
Als u de instellingen van een proportioneel Gadget wilt veranderen dan kunt u gebruik maken van de functie ModifyProp(). Het is bij deze functie niet nodig het Gadget uit de lijst te halen.

```
ModifyProp(Gadget,Wind
    ow,Requester,Flags
    ,HorizPot,VertPot,
    HorizBody,VertBody
    );
struct Gadget
    *Gadget;
struct Window
    *Window;
struct Requester
    *Requester;
```

De overige parameters zijn gelijk aan de elementen van de PropInfo-structuur.

NewModifyProp()
Identiek aan de bovenstaande functie, echter met dit verschil dat met NumGad ingesteld kan worden hoeveel Gadgets aangepast moeten worden. ModifyProp() dient dus om één proportioneel Gadget aan te passen, NewModifyProp() om meerdere proportionele Gadgets aan te passen.

```
NewModifyProp(Gadget,W
    indow,Requester,Fl
    ags,HorizPot,VertP
    ot,HorizBody,VertB
    ody,NumGad);
struct Gadget
    *Gadget;
struct Window
    *Window;
struct Requester
    *Requester;
int NumGad;
```

OffGadget()
Met de functie NumGad() is het mogelijk een Gadget 'buiten spel' te zetten. Het Gadget is na de operatie niet meer gevoelig voor aanklikken met de muis.

```
OffGadget(Gadget,Windo
    w,Requester);
struct Gadget
    *Gadget;
struct Window
    *Window;
struct Requester
    *Requester;
```

Als er geen Requester is, zal de parameter op NULL gezet moeten worden.
De flag GADGDISABLED in het element Flags van de structuur Gadget zal gezet zijn.

OnGadget()
Dit is de tegenhanger van de bovenstaande functie. Het Gadget wordt na de functie weer gevoelig voor aanklikken.

```
OnGadget(Gadget,Window
    ,Requester);
struct Gadget
    *Gadget;
struct Window
    *Window;
struct Requester
    *Requester;
```

De flag GADGDISABLED in het element flags zal na de operatie weer gewist zijn.

RefreshGadgets
Na aanroep van de functie zal het Gadget, wat met AddGadget() aan de lijst toegevoegd is, op het beeldscherm verschijnen.

```
TBPositie=RefreshGadge
    ts(Window,Gadget);

USHORT TBPositie;
struct Window
    *Window;
struct Gadget
    *Gadget;
```

Als men -1 terug krijgt in TBPositie dan is er een fout opgetreden. Naar alle waarschijnlijkheid zal de Gadget niet in de lijst staan.

RefreshGList()
In het geval van meer dan één Gadget dient u de functie RefreshGList() aan te roepen om ze op het beeldscherm te krijgen. Voorwaarde is hier ook weer dat de Gadget-lijst onderling verbonden moet zijn.

```
RefreshGList(Gadget,Wi
    ndow,Requester,Num
    Gad);
struct Gadget
    *Gadget;
struct Window
    *Window;
struct Requester
    *Requester;
int NumGad;
```

Als NumGad op -1 gezet wordt dan worden ALLE Gadgets opnieuw getekent. Bij een waarde van -2 worden alle Requester-Gadgets opnieuw getekent.

RemoveGadget()
Wanneer u een Gadget uit de lijst wilt verwijderen dient u gebruik te maken van de functie RemoveGadget().

```
TBPos=RemoveGadget(Win
    dow,Gadget)
USHORT TBPos;
struct Window
    *Window;
struct Gadget
    *Gadget;
```

Nadat u het Gadget met RemoveGadget() uit de lijst hebt gehaalt mag u veranderingen aan het Gadget uitvoeren.

RemoveGList()
Idem als hierboven alleen nu voor een gekoppelde lijst van Gadgets.

```
TBPos=RemoveGList(Wind
    ow,Gadget,NumGad)
USHORT TBPos;
struct Window
    *Window;
struct Gadget
    *Gadget;
int NumGad;
```

Het voorbeeld

Nu u op de hoogte bent van de Gadgets en hun functies kunt u het voorbeeld programma eens gaan bekijken. Typ het in, bekijk het, probeer het uit. Ga vervolgens eens wat wijzigen. Probeer zelf een Gadget toe te voegen. Hierdoor leert u meer dan wij ooit met één aflevering kunnen vertellen.
Het programma. Het programma zelf is niet meer dan een simpele verzameling van simpele Gadgets. De Gadgets zorgen ervoor dat er een signaal gegeven wordt aan het programma. Het programma reageert hierop met een specifieke actie. In het programma zijn enkele zaken verwerkt welke nog niet in de Amiga C cursus zijn behandelt. Maakt u hier zich niet te druk over. De meeste functies zijn afkomstig uit de graphics.library. Ze zorgen ervoor dat grafische data op het beeldscherm veranderd kan worden. Zo zorgt SetRGB4() ervoor dat de RGB-componenten van één kleur aangepast kan worden. Meer hierover in één van de volgende afleveringen.
Compile- en link instructies,

Aztek..

```
cc cdemo.c +L
ln cdemo.o -lc32
```

Lattice..

```
lc -L cdemo.c
```

Tot slot

Weer het einde van een aflevering. In de volgende aflevering komen de requesters en de menu's uitgebreid aan de orde waarna de Intuitionpret gebeurt is. We gaan na de volgende aflevering een niveau dieper. De graphics.library is dan aan de beurt.
Voor nu, experimenteer en probeer uit te reiken naar de grenzen van uw kunnen. Dream well!

Johan & Johan

Voorbeeld programma.

```
/* In dit programma staan een aantal
    functies afkomstig uit de
    graphics.library. Peins er niet te lang
    over, deze functies worden nog uitgebreid
    behandelt. /*
#ifndef LATTICE
 #include "functions.h"
#endif
#include "exec/types.h"
#include "graphics/gfxbase.h"
#include "intuition/intuition.h"

#define KEUZE1 1
#define KEUZE2 2
#define KEUZE3 3
#define KEUZE4 4
#define STRING 5
#define PROP1  6
#define PROP2  7


unsigned char Buffer[80]="Vul hier uw eigen
    tekst in";
unsigned char UndoBuffer[80];

SHORT Pairs[] = { 0,0,42,0,42,21,0,21,0,0 };
SHORT PairsStr[] =
    {0,0,172,0,172,10,0,10,0,0 };

struct Image ImagePr1,ImagePr2;

struct IntuiText TextStr = {
 0x01,0x00,JAM1,-51,0,NULL,
 (UBYTE *)"Invoer",NULL };
struct IntuiText TextP2 = {
 0x01,0x00,JAM1,3,-8,NULL,
 (UBYTE *)"P2",NULL };
struct IntuiText TextP1 = {
 0x01,0x00,JAM1,3,-8,NULL,
 (UBYTE *)"P1",NULL };
struct IntuiText TextK4 = {
 0x01,0x00,JAM1,16,6,NULL,
 (UBYTE *)"4",NULL };
struct IntuiText TextK3 = {
 0x01,0x00,JAM1,16,6,NULL,
 (UBYTE *)"3",NULL };
struct IntuiText TextK2 = {
 0x01,0x00,JAM1,16,6,NULL,
 (UBYTE *)"2",NULL };
struct IntuiText TextK1 = {
 0x01,0x00,JAM1,16,6,NULL,
 (UBYTE *)"1",NULL };

struct Border Border = {
 -2,-1,0x01,0x00,JAM1,5,Pairs,NULL };
struct Border BorderStr = {
 -2,-1,0x01,0x00,JAM1,5,PairsStr,NULL };

struct StringInfo StringInfo = {
 &Buffer[0],
 &UndoBuffer[0],
 0,60,0,
 NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL };
struct PropInfo P1Info = {
 AUTOKNOB|FREEVERT,
 0,0,0x0800,0x0800,
 0,0,0,0,0,0 };
struct PropInfo P2Info = {
 AUTOKNOB|FREEHORIZ,
 0,0,0x0800,0x0800,
 0,0,0,0,0,0 };

struct Gadget String = {
 NULL,60,90,170,9,GADGHCOMP,
 RELVERIFY,STRGADGET,
 (APTR)&BorderStr,NULL,&TextStr,
 NULL,(APTR)&StringInfo,7,NULL };
struct Gadget Prop2 = {
 &String,20,142,230,8,GADGHCOMP,
 GADGIMMEDIATE,PROPGADGET,
 (APTR)&ImagePr2,NULL,&TextP2,
 NULL,(APTR)&P2Info,6,NULL };
struct Gadget Prop1 = {
 &Prop2,260,50,16,100,GADGHCOMP,
 GADGIMMEDIATE,PROPGADGET,
 (APTR)&ImagePr1,NULL,&TextP1,
 NULL,(APTR)&P1Info,5,NULL };
struct Gadget Keuze4 = {
 &Prop1,200,50,40,20,GADGHCOMP,
 RELVERIFY,BOOLGADGET,
 (APRT)&Border,NULL,&TextK4,
 NULL,NULL,4,NULL };
struct Gadget Keuze3 = {
 &Keuze4,140,50,40,20,GADGHCOMP,
 RELVERIFY,BOOLGADGET,
 (APRT)&Border,NULL,&TextK3,
 NULL,NULL,3,NULL };
struct Gadget Keuze2 = {
 &Keuze3,80,50,40,20,GADGHCOMP,
 RELVERIFY,BOOLGADGET,
 (APRT)&Border,NULL,&TextK2,
 NULL,NULL,2,NULL };
struct Gadget Keuze1 = {
 &Keuze2,20,50,40,20,GADGHCOMP,
 RELVERIFY,BOOLGADGET,
 (APRT)&Border,NULL,&TextK1,
 NULL,NULL,1,NULL };
struct NewWindow NW = {

    0,0,640,240,0x00,0x01,CLOSEWINDOW|GADGETUP|
    GADGETDOWN,

    WINDOWCLOSE|SMART_REFRESH|ACTIVATE|WINDOWDR
    AG|WINDOWDEPTH,
 NULL,NULL,(UBYTE *)"Commodore INFO  C-Demo",
 NULL,NULL,0,0,0,0,WBENCHSCREEN };
struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase;
struct Window *CDemoWindow;

void GP_BailOut(f_How,f_Why)
int f_How;
char *f_Why;
{
 printf("%s",f_Why);
 exit(f_How);
}

main()
{
 struct IntuiMessage *DemoMessage;
 ULONG Class,Code,GadgetID;
 struct Gadget *GadgetPtr;
 char *Dummy;
 USHORT TBPos;

 IntuitionBase=(struct IntuitionBase *)
  OpenLibrary("intuition.library",0L);

 if(!(GfxBase=(struct GfxBase *)
  OpenLibrary("graphics.library",0L)))
   GP_BailOut(5,"Geen graphics.library\n");
```

```
if(!(CDemoWindow=(struct Window *)
 OpenWindow(&NW)))
  GP_BailOut(5,"Window wil niet openen\n");

if(!(Dummy=(char *)
 AllocMem(6,NULL)))
  GP_BailOut(5,"Geen geheugen voor Dummy\n");

TBPos=AddGList(CDemoWindow,&Keuze1,-1,7,NULL);
RefreshGList(&Keuze1,CDemoWindow,NULL,7);
ActivateGadget(&String,CDemoWindow,NULL);

SetDrMd(CDemoWindow->RPort,JAM1);

do{
 Wait(1L<<CDemoWindow->UserPort->mp_SigBit);

 while(!(DemoMessage=(struct IntuiMessage *)
   GetMsg(CDemoWindow->UserPort)));

 ReplyMsg(DemoMessage);

 Class=DemoMessage->Class;
 Code=DemoMessage->Code;
 GadgetPtr=(struct Gadget *)DemoMessage->IAddress;

 GadgetID=GadgetPtr->GadgetID;

 switch(Class)
  {
   case GADGETUP:
    switch(GadgetID)
     {
      case KEUZE1:

  SetRGB4(&CDemoWindow->WScreen->ViewPort,0,0,0,0);

       break;

      case KEUZE2:

  SetRGB4(&CDemoWindow->WScreen->ViewPort,0,15,0,0)
  ;
       break;

      case KEUZE3:

  SetRGB4(&CDemoWindow->WScreen->ViewPort,0,0,15,0)
  ;
       break;

      case KEUZE4:

  SetRGB4(&CDemoWindow->WScreen->ViewPort,0,0,0,15)
  ;
       break;

      case STRING:
       SetAPen(CDemoWindow->RPort,0);
       RectFill(CDemoWindow->RPort,5,160,630,170);

       SetAPen(CDemoWindow->RPort,2);
       Move(CDemoWindow->RPort,7,168);

  Text(CDemoWindow->RPort,Buffer,strlen(Buffer));

       SetAPen(CDemoWindow->RPort,1);
       Move(CDemoWindow->RPort,5,166);

  Text(CDemoWindow->RPort,Buffer,strlen(Buffer));
       break;
     }
   case GADGETDOWN:
    switch(GadgetID)
     {
      case PROP1:
       while(Prop1.Flags == SELECTED)
        {
         sprintf(Dummy,"%4x",P1Info.VertPot);

         SetAPen(CDemoWindow->RPort,0);

 RectFill(CDemoWindow->RPort,300,48,340,67);

         SetAPen(CDemoWindow->RPort,2);
         Move(CDemoWindow->RPort,306,56);

 Text(CDemoWindow->RPort,Dummy,strlen(Dummy));

         SetAPen(CDemoWindow->RPort,1);
         Move(CDemoWindow->RPort,302,54);

 Text(CDemoWindow->RPort,Dummy,strlen(Dummy));
        }
       break;

      case PROP2:
       while(Prop2.Flags == SELECTED)
        {
         sprintf(Dummy,"%4x",P2Info.HorizPot);

         SetAPen(CDemoWindow->RPort,0);

 RectFill(CDemoWindow->RPort,300,67,340,86);

         SetAPen(CDemoWindow->RPort,2);
         Move(CDemoWindow->RPort,306,76);

 Text(CDemoWindow->RPort,Dummy,strlen(Dummy));

         SetAPen(CDemoWindow->RPort,1);
         Move(CDemoWindow->RPort,302,74);

 Text(CDemoWindow->RPort,Dummy,strlen(Dummy));
        }
       break;
     }
  }
 }while(Class != CLOSEWINDOW);

 FreeMem(Buffer,6);
 CloseWindow(CDemoWindow);
}
```

Checksum C-64

Syntax Checksum

Het overtikken van een listing kan een heel karwei zijn en als u een beetje normaal mens bent dan maakt u daarin beslist een aantal fouten. Nu is niets moeilijker om de fouten uit je eigen werk te halen. Al geruime tijd heeft Jan Bodzinga hiervoor een zgn. Checksum-programma geschreven. Om de vele nieuwe lezers van Commodore-info te helpen volgt hieronder nog een keer een volledige uitleg over de werking van dit programma, waarmee het, hoe vreemd dat misschien ook lijkt, echt mogelijk is om met behulp van dit programma de fouten in elke door ons geplaatste listing op te sporen.

Hiervoor gaat u als volgt te werk:

1. U tikt de listing heel zorgvuldig over en SAVEt hem voordat u het programma RUNt op een diskette of cassette.
2. U tikt het RUN commando in. Mocht het programma de boodschap 'FOUT in dataregels!' geven dan heeft u een fout bij het overtikken gemaakt. Herstel dan de fout en SAVE de verbeterde versie. Mocht het programma met de boodschap 'data is weggezet checksum testen met sys...'komen dan is tot dusver alles goed. Het programma is nu in een stukje machinetaalgeheugen gezet. Als u het NEW commando geeft blijft het toch in de computer staan.

Alle door ons geplaatste programma's zijn in Basic geschreven.

Als u een programma heeft overgetikt SAVE het eerst, mocht er iets mis gaan dan hoeft u niet de gehele listing opnieuw te gaan intikken. Als u nu een programma op fouten wilt gaan controleren dan kunt u dat in het geheugen laden(wel eerst het checksum programma hebben gerund). Vervolgens typt u zonder het programma te runnen de opdracht sys 49152(c-64) of sys 1536 (c-16 en plus/4)in.

Als alles goed is gegaan loopt er nu een rij regelnummers over het scherm met getallen erachter. Dezelfde lijst staat ook achter elk door ons geplaatste programma. Wijkt nu een nummer achter een regelnummer af van het nummer dat in het blad staat dan heeft u in die regel iets anders ingetikt dan er in het blad stond. U kunt de stroom getallen d.m.v. de RUN/STOP toets pauzeren en weer vervolgen met de F1 of F7 toets. Het is uitermate belangrijk dat u goed met dit programma overweg kunt en mocht u het niet goed werkend krijgen bel dan gerust even met onze listingservice telefoonlijn. (Maandag 17.00 - 21.00 uur. Telefoonnummer 02155-25162.)

De laatste tijd wordt er weer veel gebeld zodat U nogwel eens in gesprek krijgt, daarom houdt uw vraag kort, vermeld in welk blad het desbetreffende artikel stond. Heeft U veel vragen, of zis uw vraag erg uitgebreid, doe het dan schriftelijk, zodat we veel mensen op de maandag avond te woord kunnen staan.

```
  1 rem ******************************
  2 rem basic loader "SYNTAX.CHECKSUM"
  3 rem na de commando's "run" en "new"
  4 rem blijft dit programma in het ge-
  5 rem heugen. laad het te testen pro-
  6 rem gramma en tik daarna sys 49152.
  7 rem ******************************
 10 i=49152  :rem beginadres
 20 reada:ifa<0then40:rem data ingelezen
 30 pokei,a:i=i+1:b=b+a:goto20
 40 if b<>16844thenprint"[SHIFT-CLR]fo
    ut[SPACE]in[SPACE]dataregels!":b=0:end
 50 poke49184,148:poke49185,192
 55 i=49300
 60 read a: ifa<0then80
 70 pokei,a:b=a+b:i=i+1:goto60
 80 if b<>20068thenprint"[SHIFT-CLR]fo
    ut[SPACE]in[SPACE]dataregels![SPAC
    E](vanaf[SPACE]regel[SPACE]240)":b
    =0:end
 90 print"data[SPACE]is[SPACE]weggezet"
 95 print"checksum[SPACE]testen[SPACE]
    met[SPACE]sys49152"
100 data 165,43,166,44,133,163,134,164
    ,169,147
110 data 32,210,255,160,0,240,3,32,73,192
120 data 32,73,192,208,1,96,32,225,255,208
130 data 3,76,116,164,32,81,192,32,73,192
140 data 240,12,201,32,240,247,24,101,
    167,133
150 data 167,76,37,192,166,167,169,0,1
    32,168
160 data 32,205,189,169,13,32,210,255,
    164,168
170 data 76,17,192,200,208,2,230,164,1
    77,163
180 data 96,162,0,189,123,192,240,6,32,210
190 data 255,232,208,245,32,73,192,170
    ,32,73
200 data 192,132,168,32,205,189,162,3,
    169,32
210 data 32,210,255,202,208,250,169,0,
    133,167
220 data 164,168,96,82,69,71,69,76,32,0
230 data  -1
240 data 165,197,201,3,240,7,201,4,240
250 data 6,76,148,192,76,34,192,169
260 data 147,32,210,255,76,161,192
270 data -1
```

** EINDE LISTING checksum 64 **

Checksum Checksum 64

REGEL 1	249	REGEL 120	232
REGEL 2	84	REGEL 130	183
REGEL 3	105	REGEL 140	96
REGEL 4	2	REGEL 150	96
REGEL 5	246	REGEL 160	127
REGEL 6	152	REGEL 170	71
REGEL 7	249	REGEL 180	223
REGEL 10	157	REGEL 190	73
REGEL 20	64	REGEL 200	79
REGEL 30	38	REGEL 210	109
REGEL 40	57	REGEL 220	106
REGEL 50	14	REGEL 230	225
REGEL 55	251	REGEL 240	16
REGEL 60	192	REGEL 250	163
REGEL 70	42	REGEL 260	92
REGEL 80	244	REGEL 270	22
REGEL 90	245		
REGEL 95	237		
REGEL 100	183		
REGEL 110	158		

PRINT OUT C-64 met o.a. Solitair

Balsommen

Bergen en een luchtballon, dit zijn de elementen waar mee moet worden gewerkt. De ballon verliest langzaam hoogte, hij dreigt te pletter te storten. Dit te pletter slaan is alleen te voorkomen door de gaskraan open te draaien. Er is echter een probleem, dit kan alleen gebeuren als je de som welke op het beeld verschijnt goed hebt opgelost. Dus het antwoord intikken en op return drukken, de ballon zal dan vanzelf stijgen. Er kan gewerkt worden op verschillende moeilijkheidsgraden. Het programma werkt **NIET** (goed met een ingeschakelde cartridge!! Dit programma is van F. Reijsbergen uit Leidschendam.

```
1    rem ballonsommen / cbm-64
2    rem door fons reijsbergen
3    rem leidschendam
4    rem
10   rem******start programma
20   poke52,48:poke56,48:clr
30   def fnr(x)=int(rnd(1)*x+1)
40   goto800
50   rem******setup
60   poke657,128:poke53272,22:poke53280
     ,14:poke53281,14
70   print"[SHIFT-CLR][CTRL-2]*********
     ***[SPACE]BALLONSOMMEN[SPACE]*****
     ********"
80   print"[CTRL-7][CRSR-DOWN]Over[SPAC
     E]de[SPACE]bergen[SPACE]vlieg[SPAC
     E]jij[SPACE]in[SPACE]een[SPACE]bal
     lon,[SPACE]door[SPACE]een[SPACE]ga
     t[SPACE]in[SPACE]de[SPACE]ballon[S
     PACE]";
90   print"verlies[SPACE]je[SPACE]gaszo
     dat[SPACE]de[SPACE]ballon[SPACE]st
     eeds[SPACE]lager[SPACE]komt."
100  print"Maar[SPACE]je[SPACE]kan[SPAC
     E]meer[SPACE]gas[SPACE]in[SPACE]de
     [SPACE]ballon[SPACE]laten[SPACE]lo
     pen[SPACE]door[SPACE]de[SPACE]gask
     raan";
110  print"[SPACE]open[SPACE]te[SPACE]d
     raaien."
120  print"Het[SPACE]probleem[SPACE]is[
     SPACE]de[SPACE]gaskraan,[SPACE]die
     [SPACE]werkt[2xSPACE]alleen[SPACE]
     als[SPACE]je[SPACE]eerst[SPACE]een";
130  print"[SPACE]som[SPACE]goed[SPACE]
     oplost."
140  print"Wat[SPACE]moet[SPACE]je[SPAC
     E]dus[SPACE]doen:"
150  print"Type[SPACE]de[SPACE]uitkomst
     [SPACE]van[SPACE]de[SPACE]som[SPAC
     E]in[SPACE]en[SPACE]druk[2xSPACE]o
     p[SPACE]RETURN[SPACE]om[SPACE]de[S
     PACE]kraan[SPACE]te[SPACE]";
160  print"openen.":print"Was[SPACE]de[
     SPACE]uitkomst[SPACE]goed[SPACE]da
     n[SPACE]zal[SPACE]de[SPACE]ballon[
     2xSPACE]weer[SPACE]stijgen";
170  print",[SPACE]anders[SPACE]blijft[
     SPACE]hij[SPACE]dalen."
180  print"[CRSR-DOWN]Als[SPACE]je[SPAC
     E]een[SPACE]fout[SPACE]maakt[SPACE
     ]bij[SPACE]het[SPACE]typen[SPACE]d
     rukdan[SPACE]op[SPACE]INST.DEL."
190  print"Het[SPACE]antwoord[SPACE]op[
     SPACE]een[SPACE]deling[SPACE]moet[
     SPACE]worden[2xSPACE]afgerond[SPAC
     E]op[SPACE]TWEE[SPACE]cijfers";
200  print"[SPACE]achter[SPACE]de[SPACE
     ]komma"
210  print"[CTRL-1][CRSR-DOWN][4xSPACE]
     [8xSHIFT-*][SPACE]WACHT[SHIFT-SPAC
     E]NU[SHIFT-SPACE]EVEN[SPACE][8xSHI
     FT *][CRSR-UP]"
220  poke56334,peek(56334)and254:poke1,
     peek(1)and251
230  dima$(17):b$="[CRSR-DOWN][2xCRSR-L
     EFT]":x=1
240  fori=832to895:reada:pokei,a:next
250  fori=0to511:pokei+12288,peek(i+532
     48):next
260  fori=191to222step4
270  a$(x)=a$(x)+chr$(i)+chr$(i+1)+b$+c
     hr$(i+2)+chr$(i+3)+"[CRSR-UP]":x=x
     +1:next
280  fori=159to190step4
290  a$(x)=a$(x)+chr$(i)+chr$(i+1)+b$+c
     hr$(i+2)+chr$(i+3)+"[CRSR-UP]":x=x
     +1:next
300  a$(17)="[2xSPACE][CRSR-DOWN][2xCRS
     R-LEFT][2xSPACE][CRSR-UP]"
310  a$(1)=chr$(63)+right$(a$(1),7):a$(
     9)=chr$(223)+right$(a$(9),7)
320  a$(6)="[SPACE]"+a$(6)+"[SPACE]":a$
     (8)=left$(a$(8),6)+chr$(255)+"[CRS
     R-UP]"
330  x=12288:fori=63to123step4:readc:c=
     c*8+x:p=i*8+x:p1=p
340  forj=ctoc+7:a=peek(j):b=0:ifa=0the
     n370
350  fork=0to7:j0=2~k:if(a and j0)=j0th
     enb=b+3*(j0~2)
360  nextk
370  b1=int(b/256):b2=b-256*b1
380  pokep,b1:pokep+1,b1:pokep+8,b2:pok
     ep+9,b2:p=p+2:ifp=p1+8thenp=p1+16
390  nextj,i:p=12288+29*8:fori=ptop+15:
     reada:pokei,a:next:x=32768:a=49152
400  forj=1to3:readx$:fori=1tolen(x$):p
     okex,(asc(mid$(x$,i,1))+143):x=x+1
     :nexti,j
410  readx$:fori=1tolen(x$)/2:x=10*(asc
     (mid$(x$,i*2-1))-65)+asc(mid$(x$,i
     *2))-48
420  pokea,x:a=a+1:next:ifa<49267goto410
430  poke49207,80
440  poke1,peek(1)or4:poke56334,peek(56
     334)or1
450  v=53248:poke2040,13:sh=80:sp=0.2:p
     okev,100:pokev+1,sh:pokev+39,6
460  si=54272:fori=sitosi+23:pokei,0:ne
     xt:pokesi+24,15
470  return
480  rem*****verzin + print som
490  fori=2to4:poke781,i:sys59903:next:
     i=peek(v+31):gosub780
500  g1=fnr(m1):g2=fnr(m1):so=fnr(m2):i
     fg2>g1theng=g1:g1=g2:g2=g
510  onsogoto520,530,540,550
520  ui=g1+g2:goto560
```

```
530 ui=g1-g2:goto560
540 ui=g1*g2:goto560
550 ui=(int((g1/g2)*100))/100
560 print"[HOME][CTRL-2][2xCRSR-DOWN]"
    tab(7);:ifm1<10thenprinttab(12);
570 g$=str$(g1):gosub730:print"[CTRL-2
    ][SPACE]"a$(so)"[SPACE]";:g$=str$(
    g2):gosub730:printa$(6)"[CTRL-1]";
580 rem*****invoer uitkomst
590 hit=0:fa=0:i$="":poke198,0
600 ss=ss+1:ifss>2thenss=0:sys49160
610 sh=sh+sp:ifop>0thenop=op-1:sh=sh-s
    p*2:ifsh<80thensh=80
620 pokev+1,sh:ifpeek(v+31)and1=1thenh
    it=1:return
630 getg$:if(g$<"0"org$>"9")andg$<>"."
    andg$<>chr$(13)andg$<>chr$(20)then600
640 ifg$=chr$(20)then690
650 ifg$=chr$(13)then710
660 iflen(i$)>5then630
670 ifg$="."theni$=i$+g$:printa$(5);:g
    oto630
680 i$=i$+g$:g$="[SPACE]"+g$:gosub730:
    goto600
690 ifi$=""thengoto630
700 i$=left$(i$,len(i$)-1):print"[2xCR
    SR-LEFT]"a$(17)"[2xCRSR-LEFT]";:go
    to630
710 ifi$=""then630
715 ifval(i$)<>uithenfa=1
720 return
730 fori=2tolen(g$):printa$(asc(mid$(g
    $,i,1))-41);:next:return
740 rem*****fout toon
750 pokesi+5,136:pokesi+6,240:pokesi+4
    ,17:fork=1to6:forz=250to0step-20
760 pokesi+1,z:nextz,k:pokesi+4,16:ret
    urn
770 rem*****goed toon
780 pokesi+5,8:pokesi+6,251:pokesi,0:p
    okesi+1,90:pokesi+4,17:pokesi+4,16
    :return
790 rem*****hoofdprogramma*******
800 x=rnd(-ti):gosub60:gosub780
810 print"[10xSPACE]DRUK[SPACE]OP[SHIF
    T SPACE]EEN[SHIFT-SPACE]TOETS[8xSP
    ACE]":wait197,191
820 tg=0:tf=0:l=1
830 print"[SHIFT-CLR][CTRL-2][6xCRSR-D
    OWN][2xSPACE]geef[SPACE]de[SPACE]m
    oeilijkheids[SPACE]graad[2xSPACE](
    1-4)"
840 geta$:ifa$<"1"ora$>"4"then840
850 m2=val(a$):m1=9:aa=0:print"[SHIFT-
    CLR][COM-8][4xSPACE]level[SPACE]0[
    4xSPACE]goed[SPACE]0[5xSPACE]fout[
    SPACE]0[CTRL-1]"
860 pokev+21,1:poke53272,28:sys49152
870 print"[HOME]"tab(10)ltab(20)tgtab(
    31)tf
880 gosub490:ifhit=1then1070
890 iffa=0thentg=tg+1:op=5/sp:goto970
900 rem*****fout antwoord
910 ifi$=""thenprint"[2xCRSR-RIGHT]";
920 fori=1tolen(i$):print"[2xCRSR-LEFT
    ]"a$(17)"[2xCRSR-LEFT]";:next:prin
    t"[CTRL-7][SPACE]";:g$=str$(ui):go
    sub730
930 print"[CTRL-1]";:gosub750:fori=1to
    50:ss=ss+0.5:ifss>2thenss=0:sys49160
940 ifpeek(v+31)and1=1then1070
950 sh=sh+sp/2:pokev+1,sh:next
960 op=0:tf=tf+1
970 aa=aa+1:ifaa=10then1000
980 goto870
990 rem*****volgend level
1000 l=l+1:aa=0:m1=m1+10:ifm1>99thenm1=
     9:sp=sp+0.2
1010 print"[HOME][9xCRSR-DOWN][6xSPACE]
     goed[SPACE]zo.[SPACE]nu[SPACE]op[S
     PACE]naar[SPACE]level"l
1020 gosub780:fori=shto80step-.1:pokev+
     1,i:next:sh=80:gosub780
1030 print"[2xCRSR-DOWN][10xSPACE]druk[
     SPACE]op[SPACE]een[SPACE]toets[8xS
     PACE]":wait197,191:gosub780
1040 poke781,9:sys59903:poke781,12:sys5
     9903
1050 goto870
1060 rem****op bergen gelopen
1070 pokesi+6,250:pokesi,160:pokesi+24,31
1080 pokes+13,254:pokesi+8,40:pokesi+4,
     129:pokesi+11,129
1090 fori=1to150:poke832+fnr(64),0:next
1100 pokev+21,0:pokesi+4,128:pokesi+11,129
1110 print"[HOME][9xCRSR-DOWN][2xSPACE]
     jammer[SPACE]maar[SPACE]je[SPACE]b
     allon[SPACE]is[SPACE]neergestort"
1120 l=(tg-tf)*l:ifl<0thenl=1
1130 printtab(12)"[3xCRSR-DOWN]je[SPACE
     ]score[SPACE]is[SPACE]"l
1140 print"[3xCRSR-DOWN][5xSPACE]wil[SP
     ACE]je[SPACE]het[SPACE]nog[SPACE]e
     ens[SPACE]probeeren"
1150 geta$:ifa$=""then1150
1160 ifa$="n"thensys64738
1170 ifa$<>"j"then1150
1180 restore:fori=832to895:reada:pokei,
     a:next:sh=80:sp=0.2:pokev+1,sh
1190 goto820
1200 rem****data
1210 data0,127,0,1,255,192,3,255,224,3,
     231,224,7,217,240,7,223,240,7,217,
     240,3
1220 data231,224,3,255,224,3,255,224,2,
     255,160,1,127,64,1,62,64,0,156,128,0
1230 data156,128,0,73,0,0,73,0,0,62,0,0
     ,62,0,0,62,0,0,28,0,0
1240 :
1250 data43,45,24,58,46,61,48,49,50,51,
     52,53,54,55,56,57
1260 :
1270 data192,96,48,24,12,12,6,3,3,6,6,1
     2,56,96,96,192
1280 :
1290 data xwghxhxwvfgwghxwgwvfghxhxwgwv
     uefvfgwvfvuefgwvfghxwvutdef
1300 data gwvuefvfghxwvuefvutscdefgwvfg
     hxwvutscdeutsrbcdtscdefvutscs
1310 data rbcdtsrqabcsrbcsrqabrbcdeutsr
     qabrbcdtscdeutdtsrqabcdefghxh
1320 :
1330 data q0a0n2z0n2z1n2z2f6d2y0z5p2h2n
     8h2d2k0t2q4z2s5a0m8u0t2r2u8a2q0a0n
     2z2q4
1340 data z0h2e1b5c4k5b6r0c4d2y0z5k4e1y
```

```
     0h4h4h4h4c4k5t2d2v0z5u0t2d9u8b8q4z
     1u0t2
1350 data r1u8a2q0a0n2z1n2z2h6i7t2q0a0n
     2z0h6b9t2q0a0n2z0k4r0k4q8c4d2y0z5j
     6q2a0
1360 data q9d2p7y0a5p7y0a6x2u8y7j6x4
```

** EINDE LISTING balsom **

Checksum balsom

REGEL 1	174	REGEL 700	178
REGEL 2	41	REGEL 710	46
REGEL 3	36	REGEL 715	33
REGEL 4	143	REGEL 720	142
REGEL 10	191	REGEL 730	152
REGEL 20	64	REGEL 740	223
REGEL 30	10	REGEL 750	113
REGEL 40	33	REGEL 760	199
REGEL 50	28	REGEL 770	192
REGEL 60	45	REGEL 780	178
REGEL 70	22	REGEL 790	157
REGEL 80	167	REGEL 800	241
REGEL 90	192	REGEL 810	141
REGEL 100	245	REGEL 820	156
REGEL 110	202	REGEL 830	126
REGEL 120	66	REGEL 840	217
REGEL 130	250	REGEL 850	39
REGEL 140	89	REGEL 860	158
REGEL 150	74	REGEL 870	252
REGEL 160	247	REGEL 880	38
REGEL 170	238	REGEL 890	209
REGEL 180	62	REGEL 900	13
REGEL 190	53	REGEL 910	231
REGEL 200	146	REGEL 920	146
REGEL 210	23	REGEL 930	201
REGEL 220	227	REGEL 940	52
REGEL 230	250	REGEL 950	153
REGEL 240	168	REGEL 960	124
REGEL 250	164	REGEL 970	83
REGEL 260	46	REGEL 980	40
REGEL 270	80	REGEL 990	232
REGEL 280	54	REGEL 1000	152
REGEL 290	80	REGEL 1010	167
REGEL 300	240	REGEL 1020	196
REGEL 310	200	REGEL 1030	213
REGEL 320	134	REGEL 1040	96
REGEL 330	201	REGEL 1050	40
REGEL 340	153	REGEL 1060	147
REGEL 350	215	REGEL 1070	19
REGEL 360	205	REGEL 1080	242
REGEL 370	192	REGEL 1090	201
REGEL 380	53	REGEL 1100	169
REGEL 390	74	REGEL 1110	155
REGEL 400	55	REGEL 1120	241
REGEL 410	84	REGEL 1130	50
REGEL 420	68	REGEL 1140	29
REGEL 430	49	REGEL 1150	148
REGEL 440	23	REGEL 1160	133
REGEL 450	143	REGEL 1170	80
REGEL 460	52	REGEL 1180	69
REGEL 470	142	REGEL 1190	35
REGEL 480	230	REGEL 1200	81
REGEL 490	28	REGEL 1210	175
REGEL 500	120	REGEL 1220	35
REGEL 510	162	REGEL 1230	16
REGEL 520	73	REGEL 1240	58
REGEL 530	74	REGEL 1250	168
REGEL 540	75	REGEL 1260	58
REGEL 550	17	REGEL 1270	10
REGEL 560	86	REGEL 1280	58
REGEL 570	18	REGEL 1290	198
REGEL 580	180	REGEL 1300	25
REGEL 590	214	REGEL 1310	92
REGEL 600	50	REGEL 1320	58
REGEL 610	18	REGEL 1330	180
REGEL 620	197	REGEL 1340	138
REGEL 630	28	REGEL 1350	140
REGEL 640	104	REGEL 1360	630
REGEL 650	99		
REGEL 660	50		
REGEL 670	183		
REGEL 680	11		
REGEL 690	183		

sterren

Een soort speelbord met een aantal sterren erop. Het spel dat hierop gespeeld wordt bestaat uit twee gedeeltes. Er moet diagonaal over de sterren worden heengesprongen. In het eerste deel is de bedoeling dat er zo veel mogelijk sterren blijven staan en in deel twee is de opzet dat er zo weinig mogelijk sterren overblijven. Ook dit spel is er een van f. Reijsbergen.

```
1    rem sterren slaan / cbm-64
2    rem door fons reijsbergen
3    rem
4    rem
10   poke53280,0:poke53281,0:poke53272,
     23
20   goto900
30   rem*******setup
40   restore:fori=1to8:forj=1to8:a(i,j)
     =0:nextj,i
50   fori=1to24:reada:a(a/10,int(.5+((a
     /10)-int(a/10))*10))=1:next
60   fori=0to8:a(0,i)=8:a(9,i)=8:a(i,0)
     =8:a(i,9)=8:next:return
70   rem*******variable
80   t$="[7xSPACE]":fase=1:s1=24:s2=24
90   s$(1)=t$+"[SPACE]A[SPACE]B[SPACE]C
     [SPACE]D[SPACE]E[SPACE]F[SPACE]G[S
     PACE]H"+chr$(13)+t$+"[COM-A][SHIFT
      *][COM-R][SHIFT-*][COM-R][SHIFT-*
     ][COM-R][SHIFT-*][COM-R][SHIFT-*][
     COM-R][SHIFT-*][COM-R][SHIFT-*][CO
     M R][SHIFT-*][COM-S]"
100  s$(2)=t$+"[SHIFT--][SPACE][SHIFT--
     ][SPACE][SHIFT--][SPACE][SHIFT--][
     SPACE][SHIFT--][SPACE][SHIFT--][SP
     ACE][SHIFT--][SPACE][SHIFT--][SPAC
     E][SHIFT--]"
110  s$(3)=t$+"[COM-Q][SHIFT-*][SHIFT-+
     ][SHIFT-*][SHIFT-+][SHIFT-*][SHIFT
      +][SHIFT-*][SHIFT-+][SHIFT-*][SHI
     FT +][SHIFT-*][SHIFT-+][SHIFT-*][S
     HIFT +][SHIFT-*][COM-W]"
120  s$(4)=t$+"[COM-Z][SHIFT-*][COM-E][
     SHIFT-*][COM-E][SHIFT-*][COM-E][SH
     IFT *][COM-E][SHIFT-*][COM-E][SHIF
     T *][COM-E][SHIFT-*][COM-E][SHIFT-
     *][COM-X]"
130  s$(5)="FASE[SPACE]1[SPACE]Hou[SPAC
     E]zoveel[SPACE]mogelijk[SPACE]ster
     ren[SPACE]over."
140  s$(6)="FASE[SPACE]2[SPACE]Maak[SPA
     CE]zoveel[SPACE]mogelijk[SPACE]spr
     ongen.[3xSPACE]"
150  s$(7)="[40xCOM-T]"
160  s$(8)="[HOME][22xCRSR-DOWN]"
170  return
180  rem******print speelveld
190  print"[SHIFT-CLR]"
200  print"[HOME][2xSPACE]*********[SPA
     CE]Sterren[SPACE]Slaan[SPACE]*****
     ****[CRSR-DOWN]"
210  prints$(1):fori=1to7:prints$(2):pr
     ints$(3):next:prints$(2):print s$(
     4)
220  print"[HOME][3xCRSR-DOWN]":fori=1t
```

```
    o8:printtab(4)i"[CRSR-DOWN]":next
230 iffase=1thenprints$(5)s$(7)
240 iffase=2thenprints$(6)s$(7)
250 print"[HOME][3xCRSR-DOWN]"tab(28)"
    PUNTEN"
260 print"[3xCRSR-DOWN]"tab(25)"fase[S
    PACE]1[SPACE]:"
270 print"[CRSR-DOWN]"tab(25)"fase[SPA
    CE]2[SPACE]:"
280 print"[CRSR-DOWN]"tab(33)"[4xSHIFT
     *][SPACE]-"
290 print"[CRSR-DOWN]"tab(25)"score[2x
    SPACE]:"
300 rem******print stukken
310 print"[HOME][4xCRSR-DOWN]"tab(8);:
    fori=1to8:forj=1to8:ifa(i,j)thenpr
    int"*[CRSR-RIGHT]";:goto330
320 print"[SPACE][CRSR-RIGHT]";
330 next:print:print"[CRSR-DOWN]"tab(8
    );:next
340 rem******print score
350 print"[HOME][7xCRSR-DOWN]"tab(33);
    s1"[CRSR-LEFT][SPACE]":print"[CRSR
    -DOWN]"tab(33);s2"[CRSR-LEFT][SPAC
    E]"
360 print"[3xCRSR-DOWN]"tab(33)"[SPACE
    ][CRSR-LEFT]"s1-s2"[CRSR-LEFT][SPA
    CE]"
370 return
380 rem******doe een zet
390 prints$(8)"ster[SPACE]van[SPACE]";
    :poke204,0:gosub650:poke204,1:prin
    t"[SPACE][CRSR-LEFT][SPACE]";
400 x1=val(j$):y1=asc(i$)-64
410 print"naar[SPACE]";:poke204,0:gosu
    b650:poke204,1
420 x2=val(j$):y2=asc(i$)-64
430 poke781,22:sys59903
440 rem******kontrolle zet
450 if(abs(x1-x2)<>2)or(abs(y1-y2)<>2)
    then620
460 x3=x1-1:ifx1<x2thenx3=x1+1
470 y3=y1-1:ify1<y2theny3=y1+1
480 ifa(x1,y1)<>1 or a(x2,y2)<>0or a(
    x3,y3)<>1then620
490 rem******voer zet uit
500 a(x1,y1)=0:a(x2,y2)=1:a(x3,y3)=0
510 iffase=1thens1=s1-1
520 iffase=2thens2=s2-1
530 rem******nog zet mogelijk?
540 zm=0:fori=1to8:forj=1to8
550 ifi<>1andj<>1thenif(a(i,j)+a(i-1,j
    -1)=2)anda(i-2,j-2)=0then600
560 ifi<>1thenif(a(i,j)+a(i-1,j+1)=2)a
    nda(i-2,j+2)=0then600
570 ifj<>1thenif(a(i,j)+a(i+1,j-1)=2)a
    nda(i+2,j-2)=0then600
580 if(a(i,j)+a(i+1,j+1)=2)anda(i+2,j+
    2)=0then600
590 nextj,i:return
600 zm=1:i=8:j=8:goto590
610 rem******invoer fout
620 prints$(8)"Je[SPACE]hebt[SPACE]een
    [SPACE]invoer[SPACE]fout[SPACE]gem
    aakt."
630 wait203,64:wait197,191:poke781,22:
    sys59903:goto390
640 rem******input
650 geta$:ifa$<"a"ora$>"h"then650
660 printa$;:i$=a$
670 geta$:if(a$<"1"ora$>"8")anda$<>chr
    $(20)then670
680 ifa$=chr$(20)thenprint"[SPACE][2xC
    RSR-LEFT][SPACE][CRSR-LEFT]";:goto
    650
690 printa$;:j$=a$
700 geta$:ifa$<>chr$(13)anda$<>chr$(20
    )then700
710 ifa$=chr$(20)thenprint"[SPACE][2xC
    RSR-LEFT][SPACE][CRSR-LEFT]";:goto
    670
720 return
730 rem*******spel uitleg
740 print"[SHIFT-CLR][2xSPACE]********
    *[SPACE]STERREN[SPACE]SLAAN[SPACE]
    *********"
750 print"[2xCRSR-DOWN]Bij[SPACE]dit[S
    PACE]spel[SPACE]moet[SPACE]je[SPAC
    E]door[SPACE]herhaaldelijk"
760 print"springen[SPACE]de[SPACE]ster
    ren[SPACE]van[SPACE]het[SPACE]spee
    lbord[3xSPACE]verwijderen."
770 print"Je[SPACE]moet[SPACE]met[SPAC
    E]een[SPACE]ster[SPACE]diagonaal[S
    PACE]over[SPACE]een"
780 print"andere[SPACE]ster[SPACE]spri
    ngen[SPACE]naar[SPACE]een[SPACE]le
    eg[SPACE]vak."
790 print"De[SPACE]ster[SPACE]wordt[SP
    ACE]dan[SPACE]weg[SPACE]gehaalt."
800 print"[CRSR-DOWN][4xSPACE]Het[SPAC
    E]spel[SPACE]bestaat[SPACE]uit[SPA
    CE]twee[SPACE]fasen."
810 print"[CRSR-DOWN]Bij[SPACE]FASE[SP
    ACE]1[SPACE]is[SPACE]het[SPACE]de[
    SPACE]bedoeling[SPACE]om[SPACE]zov
    eel";
820 print"mogelijk[SPACE]sterren[SPACE
    ]te[SPACE]laten[SPACE]staan."
830 print"[CRSR-DOWN]Bij[SPACE]FASE[SP
    ACE]2[SPACE]is[SPACE]het[SPACE]de[
    SPACE]bedoeling[SPACE]om[SPACE]zov
    eel";
840 print"mogelijk[SPACE]sterren[SPACE
    ]weg[SPACE]te[SPACE]halen."
850 print"[CRSR-DOWN][3xSPACE]De[SPACE
    ]score[SPACE]is[SPACE]het[SPACE]aa
    ntal[SPACE]sterren[SPACE]van"
860 print"Fase[SPACE]1[SPACE]min[SPACE
    ]het[SPACE]aantal[SPACE]sterren[SP
    ACE]van[SPACE]Fase[SPACE]2"
870 print"[2xCRSR-DOWN][3xSPACE]******
    **DRUK[SPACE]OP[SHIFT-SPACE]EEN[SP
    ACE]TOETS********"
880 wait203,64:wait197,191:print"[SHIF
    T CLR]":return
890 rem******hoofdloop
900 gosub740:gosub40:gosub80:gosub190
910 gosub390:gosub310:ifzm=1then910
920 fase=2:gosub40:gosub200
930 gosub390:gosub310:ifzm=1then930
940 s1=s1-s2:prints$(8)"[2xCRSR-UP]Het
    [SPACE]spel[SPACE]is[SPACE]afgelop
    en.[SPACE]De[SPACE]score[SPACE]is[
    SPACE]"s1"[CRSR-DOWN]"
950 ifs1=15thenprint"DAT[SPACE]IS[SPAC
```

```
    E]REDELIJK."
960 ifs1=16thenprint"DAT[SPACE]IS[SPAC
    E]ZEER[SPACE]GOED."
970 ifs1=17thenprint"DAT[SPACE]IS[SPAC
    E]UITSTEKEND."
980 print"[11xSPACE]Nog[SPACE]een[SPAC
    E]poging[SPACE]?"
990 geta$:ifa$="j"thenrun
1000 goto990
1010 rem*****data  plaats sterren
1020 data11,13,15,17,22,24,26,28,31,37,
    42,48
1030 data51,57,62,68,71,73,75,77,82,84,
    86,88
```

** EINDE LISTING sterren **

Checksum sterren

REGEL 1	185	REGEL 520	205
REGEL 2	41	REGEL 530	243
REGEL 3	36	REGEL 540	16
REGEL 4	143	REGEL 550	59
REGEL 10	138	REGEL 560	171
REGEL 20	34	REGEL 570	172
REGEL 30	70	REGEL 580	153
REGEL 40	251	REGEL 590	9
REGEL 50	176	REGEL 600	198
REGEL 60	138	REGEL 610	156
REGEL 70	251	REGEL 620	32
REGEL 80	87	REGEL 630	51
REGEL 90	167	REGEL 640	27
REGEL 100	215	REGEL 650	252
REGEL 110	110	REGEL 660	247
REGEL 120	85	REGEL 670	33
REGEL 130	99	REGEL 680	16
REGEL 140	160	REGEL 690	248
REGEL 150	109	REGEL 700	64
REGEL 160	127	REGEL 710	18
REGEL 170	142	REGEL 720	142
REGEL 180	188	REGEL 730	179
REGEL 190	112	REGEL 740	246
REGEL 200	135	REGEL 750	9
REGEL 210	71	REGEL 760	200
REGEL 220	19	REGEL 770	177
REGEL 230	201	REGEL 780	87
REGEL 240	203	REGEL 790	142
REGEL 250	119	REGEL 800	73
REGEL 260	17	REGEL 810	206
REGEL 270	240	REGEL 820	4
REGEL 280	145	REGEL 830	207
REGEL 290	27	REGEL 840	100
REGEL 300	61	REGEL 850	169
REGEL 310	111	REGEL 860	252
REGEL 320	53	REGEL 870	123
REGEL 330	120	REGEL 880	99
REGEL 340	148	REGEL 890	53
REGEL 350	52	REGEL 900	227
REGEL 360	184	REGEL 910	20
REGEL 370	142	REGEL 920	135
REGEL 380	46	REGEL 930	22
REGEL 390	19	REGEL 940	13
REGEL 400	206	REGEL 950	24
REGEL 410	35	REGEL 960	36
REGEL 420	208	REGEL 970	208
REGEL 430	169	REGEL 980	28
REGEL 440	56	REGEL 990	161
REGEL 450	212	REGEL 1000	43
REGEL 460	117	REGEL 1010	99
REGEL 470	123	REGEL 1020	55
REGEL 480	97	REGEL 1030	11
REGEL 490	172		
REGEL 500	148		
REGEL 510	202		

solitair

Ook dit uitstekende denkprorgamma is van de hand van F. Reijsbergen. De zomermaanden zijn bij uitstek geschikt voor het "spelen" met dit soort programma's. Een ieder kent ze wel, de solitair spelen. Dit programma kent echter zeer veel variaties. Ze variëren van moeilijk tot zeer moeilijk.

```
1   rem solitaire puzzels / cbm-64
2   rem door fons reijsbergen
3   rem leidschendam
4   rem
10  poke53280,0:poke53281,0:poke53272,
    23:poke657,128
20  s$="[SHIFT-CLR][CTRL-2]***********
    ****[SPACE]SOLITAIR[SPACE]********
    *******"
30  prints$"[CRSR-DOWN]Een[SPACE]beken
    d[SPACE]spel,[SPACE]maar[SPACE]wis
    t[SPACE]je[SPACE]dat[SPACE]er[SPAC
    E]met";
40  print"dit[SPACE]spel[SPACE]meer[SP
    ACE]puzzels[SPACE]zijn[SPACE]te[SP
    ACE]maken[SPACE]?"
50  print"[CRSR-DOWN]Voor[SPACE]alle[S
    PACE]puzzels[SPACE]geld[SPACE]het[
    SPACE]volgende:"
60  print"[CRSR-DOWN]-[SPACE]Aan[SPACE
    ]het[SPACE]eind[SPACE]van[SPACE]he
    t[SPACE]spel[SPACE]moet[SPACE]je[S
    PACE]een[SPACE]rondje[SPACE][O][SP
    ACE]over[SPACE]houden,[SPACE]";
70  print"die[SPACE]dan[SPACE]op[SPACE
    ]de[3xSPACE]plaats[SPACE]moet[SPAC
    E]staan[SPACE]die[SPACE]door[SPACE
    ]EIND[SPACE]OP[SPACE]aan-[SPACE]ge
    geven[SPACE]wordt."
80  print"-[SPACE]Probeer[SPACE]bovens
    taand[SPACE]in[SPACE]zo[SPACE]min[
    SPACE]mogelijkzetten[SPACE]te[SPAC
    E]bereiken."
90  print"-[SPACE]Een[SPACE]zet[SPACE]
    is[SPACE]het[SPACE]verplaatsen[SPA
    CE]van[SPACE]een[4xSPACE]rondje"
100 print"-[SPACE]Een[SPACE]rondje[SPA
    CE]kan[SPACE]verplaatst[SPACE]word
    en[SPACE]door[SPACE]over[SPACE]een
    [SPACE]ander[SPACE]heen[SPACE]te[S
    PACE]";
110 print"springen[SPACE]naar[SPACE]ee
    nleeg[SPACE]vakje."
120 print"-[SPACE]Het[SPACE]rondje[SPA
    CE]waar[SPACE]overheen[SPACE]gespr
    ongen[SPACE]iswordt[SPACE]van[SPAC
    E]het[SPACE]bord[SPACE]gehaald."
130 print"-[SPACE]Er[SPACE]mag[SPACE]a
    lleen[SPACE]horizontaal[SPACE]en[S
    PACE]vertikaalgesprongen[SPACE]wor
    den."
140 readaa:dimp$(aa,2),p(aa,3),o$(aa)
150 fori=1toaa:readp$(i,1),p(i,1),p(i,
    2),p(i,3),p$(i,2),o$(i),o$:o$(i)=o
    $(i)+o$
160 next:gosub900
170 prints$"[CRSR-DOWN]Het[SPACE]intyp
    en[SPACE]van[SPACE]een[SPACE]spron
    g[SPACE]gaat[SPACE]alsvolgt"
```

```
180 print"Elk[SPACE]vakje[SPACE]wordt[
    SPACE]aangegeven[SPACE]door[SPACE]
    twee[4xSPACE]tekens.[2xSPACE]Het[S
    PACE]eerste[SPACE]teken[2xSPACE]";
190 print"is[SPACE]de[SPACE]letter[SPA
    CE]";:print"van[SPACE]de[SPACE](ho
    rizontale)[SPACE]rij.[SPACE]Het[SP
    ACE]tweede"
200 print"teken[SPACE]is[SPACE]het[SPA
    CE]nummer[SPACE]van[SPACE]de[SPACE
    ](verticale)[2xSPACE]kolom."
210 print"[CRSR-DOWN]Geef[SPACE]eerst[
    SPACE]het[SPACE]vakje[SPACE]aan[SP
    ACE]waar[SPACE]het[SPACE]rondjevan
    daan[SPACE]komt[SPACE]en[SPACE]na[
    SPACE]het";
220 print"[SPACE]streepje[SPACE]waar[S
    PACE]hetrondje[SPACE]heen[SPACE]mo
    et.[SPACE]Druk[SPACE]dan[SPACE]op[
    SPACE]’RETURN’"
230 print"Met[SPACE]de[SPACE]toets[SPA
    CE]’INSTDEL’[SPACE]kun[SPACE]je[SP
    ACE]typefouten";
240 print"verbeteren."
250 print"[3xCRSR-DOWN]Dit[SPACE]progr
    amma[SPACE]heeft"aa"puzzels[SPACE]
    in[SPACE]het":print"geheugen[SPACE
    ]staan."
260 gosub900
270 v$="[COM-A][SHIFT-*][COM-S][CRSR-D
    OWN][3xCRSR-LEFT][SHIFT--][SPACE][
    SHIFT--][CRSR-DOWN][3xCRSR-LEFT][C
    OM Z][SHIFT-*][COM-X][2xCRSR-UP]":
    v1$="[COM-A][SHIFT-*][COM-S][CRSR-
    DOWN][3xCRSR-LEFT][SHIFT--]O[SHIFT
     -][CRSR-DOWN][3xCRSR-LEFT][COM-Z]
    [SHIFT-*][COM-X][2xCRSR-UP]":v2$="
    van-naar[SPACE]:"
280 prints$"[2xCRSR-DOWN]":ss=5:zt=0
290 fori=1toaa:printitab(4)p$(i,1):nex
    t:printitab(4)"Zelf[SPACE]maken"
300 poke198,0
310 printtab(4)"[CRSR-DOWN]Maak[SPACE]
    een[SPACE]keuze[SPACE]1[SPACE]-"aa
    +1;:inputi$:ifi$=""thenprint"[2xCR
    SR-UP]";:goto310
320 ifval(i$)<1orval(i$)>aa+1thenprint
    "[2xCRSR-UP]";:goto310
330 print"[SHIFT-CLR]":sp=val(i$):ifsp
    =aa+1thenss=2:gosub770:ss=5:goto36
    0
340 n$=p$(sp,1):a$=p$(sp,2):ei=p(sp,1)
    :an=p(sp,2):mi=p(sp,3)
350 fori=0to7:forj=1to7:a(i+1,j)=val(m
    id$(a$,i*7+j,1)):nextj,i
360 ei$=chr$(int(ei/10)+192)+mid$(str$
    (ei),3)
370 gosub420:print:ifs=3thenprint"[CRS
    R-UP]KLAAR[2xSPACE]";:goto410
380 ifs=-1thenprint"[CRSR-UP]GEWELDIG[
    SPACE]HET[SPACE]IS[SPACE]GELUKT.";
    :goto410
390 ifs=0thenprint"[CRSR-UP]TOCH[SPACE
    ]EEN[SPACE]GOEDE[SPACE]POGING.";:g
    oto410
400 ifs=1thenprint"[CRSR-UP]PROBEER[SP
    ACE]HET[SPACE]NOG[SPACE]EENS.";
410 print"druk[SPACE]op[SPACE]’CTRL’";
    :wait653,4:goto280
420 print"[SHIFT-CLR]"n$:print"[4xCRSR
    -DOWN]Zet":print"[3xCRSR-DOWN]Min"
    mi:print"[2xCRSR-DOWN]Eind[SPACE]o
    p[SPACE]"ei$
430 print"[3xCRSR-DOWN]S=Stop":print"[
    CRSR-DOWN]O=Oplossing"
440 gosub560
450 gosub620:ifs=1ors=3thenreturn
460 ifv1<>n1andv2<>n2thengoto550
470 ifv1=n1thenm1=v1:m2=(v2+n2)/2:ifab
    s(v2-n2)<>2then550
480 ifv2=n2thenm2=v2:m1=(v1+n1)/2:ifab
    s(v1-n1)<>2then550
490 if(a(v1,v2)<>1)or(a(n1,n2)<>0)or(a
    (m1,m2)<>1)then550
500 zt=zt+1
510 a(v1,v2)=0:a(n1,n2)=1:a(m1,m2)=0:a
    n=an-1:ifan=1then530
520 goto440
530 gosub560:ifn1*10+n2=eithens=-1
540 return
550 print:print"[CRSR-UP]je[SPACE]hebt
    [SPACE]een[SPACE]fout[SPACE]gemaak
    t";:wait197,191:goto450
560 print"[HOME][5xCRSR-DOWN][3xCRSR-R
    IGHT]"zt
570 print"[HOME]"tab(15)"[CRSR-DOWN][S
    PACE]A[2xSPACE]B[2xSPACE]C[2xSPACE
    ]D[2xSPACE]E[2xSPACE]F[2xSPACE]G":
    printtab(15);
580 fori=1to7:forj=1to7:ifa(i,j)=2then
    print"[3xSPACE]";
590 ifa(i,j)=1thenprintv1$;
600 ifa(i,j)=0thenprintv$;
610 next:print"[CRSR-DOWN]"i:printtab(
    15)"[CRSR-DOWN]";:next:return
620 print:poke781,23:sys59903:print"[C
    RSR-UP]"v2$;:i$="":s=0
630 geta$:ifa$=""then630
640 ifa$="o"thenprint"wacht";:goto920
650 ifa$="s"thens=1:return
660 ifa$=chr$(13)andlen(i$)=ssthen740
670 ifa$=chr$(20)andi$<>""thenprint"[C
    RSR-LEFT][SPACE][CRSR-LEFT]";:i$=l
    eft$(i$,len(i$)-1):goto630
680 if(len(i$)=1orlen(i$)=4)and(a$<"0"
    ora$>"7")then630
690 if(len(i$)=0orlen(i$)=3)and(a$<"a"
    ora$>"g")then630
700 iflen(i$)=ssthen630
710 aa$=a$:ifa$>"’"theni=asc(a$)or128:
    a$=chr$(i):aa$=mid$(str$(i-192),2)
720 printa$;:i$=i$+aa$:iflen(i$)=2ands
    s=5theni$=i$+"-":print"-";
730 goto630
740 i$=i$+"00":v2=val(mid$(i$,1,1)):v1
    =val(mid$(i$,2,1))
750 n2=val(mid$(i$,4,1)):n1=val(mid$(i
    $,5,1))
760 return
770 a(1,1)=2:a(1,2)=2:a(1,6)=2:a(1,7)=
    2:a(2,1)=2:a(2,7)=2
780 a(7,1)=2:a(7,2)=2:a(7,6)=2:a(7,7)=
    2:a(6,1)=2:a(6,7)=2
790 n$="gebruiker":an=0:ei=44
800 print"[SHIFT-CLR]"n$:print"[4xCRSR
    -DOWN]zet":print"[3xCRSR-DOWN]Geef
```

```
     [SPACE]de[SPACE]plaats":print"aan[
     SPACE]waar[SPACE]een"
810  print"rondje[SPACE]moet":print"kom
     en[SPACE]of[SPACE]waar":print"het[
     SPACE]rondje[SPACE]weg"
820  print"gehaald[SPACE]moet":print"wo
     rden."
830  print"[3xCRSR-DOWN]S=Stop"
840  gosub560
850  print:poke781,23:sys59903:print"[C
     RSR-UP]plaats[SPACE]:";:i$="":s=0:
     gosub630
860  ifs=1thenreturn
870  ifa(v1,v2)=2then850
880  ifa(v1,v2)=0thena(v1,v2)=1:an=an+1
     :goto840
890  ifa(v1,v2)=1thena(v1,v2)=0:an=an-1
     :goto840
900  print"[CRSR-DOWN][4xSPACE]******dr
     uk[SPACE]nu[SPACE]op[SPACE]een[SPA
     CE]toets*****"
910  wait197,191:poke198,0:return
920  ifsp=17thens=0:return
930  a$=p$(sp,2):fori=0to7:forj=1to7:a(
     i+1,j)=val(mid$(a$,i*7+j,1)):nextj
     ,i
940  zt=0:forx=1tolen(o$(sp))step4:gosu
     b560
950  v2=val(mid$(o$(sp),x,1)):v1=val(mi
     d$(o$(sp),x+1,1))
960  n2=val(mid$(o$(sp),x+2,1)):n1=val(
     mid$(o$(sp),x+3,1))
970  m1=v1:m2=(v2+n2)/2:ifv2=n2thenm2=v
     2:m1=(v1+n1)/2
980  a(v1,v2)=0:a(n1,n2)=1:a(m1,m2)=0:z
     t=zt+1
990  i$=chr$(v2+192)+mid$(o$(sp),x+1,1)
     +"-"+chr$(n2+192)+mid$(o$(sp),x+3,
     1)
1000 print:print"[CRSR-UP]"v2$i$;
1010 rem wachtlus
1020 next:gosub560:s=3:return
1028 :
1029 :
1030 data16,"Latijns[SPACE]kruis"
1040 data44,6,5,2200022200100200111000
     0100000010002000002220022
1050 data432345435333,23434244
1059 :
1060 data"Grieks[SPACE]kruis"
1070 data44,9,8,2200022200100200010000 1
     1111000010002001002220022
1080 data4547434564444446,2444474545434
     244
1089 :
1090 data"Voetbalelftal[SPACE]1"
1100 data44,11,10,2211122201110200111 00
     0010100000000020000022200022
1110 data4222436334325153,5452315151536
     34322424244
1119 :
1120 data"Voetbalelftal[SPACE]2"
1130 data44,11,10,2200022220002200000 00
     0010100001110022111222211122
1140 data4525373534365737,3735254546444
     46456546444
1149 :
1150 data"Piramide[SPACE]van[SPACE]Chef
     ren"
1160 data44,9,8,2200022220002200010000 0
     1110001111022000222200022
1170 data5553533335556545,4446333525454
     644
1179 :
1180 data"Grafteken"
1190 data43,15,14,2201022200100201111 10
     0001000000100020111022211122
1200 data37355755456547453555 6545,44464
     2446343434523434644444 24143
1209 :
1210 data"Piramide[SPACE]van[SPACE]Cheo
     ps"
1220 data44,16,15,2200022200100200111 00
     0111110111111200000222 00022
1230 data3436153554567555363434545452 42
     4456545434,2444454333535 2546444
1239 :
1240 data"Schemerlamp"
1250 data44,17,16,2201022201110201111 10
     0001000000100020111022211122
1260 data3234525457555353533335363434 7
     4537353555
1270 data5553533341434442234342 44
1279 :
1280 data"Klassiek"
1290 data44,32,31,2211122221112211111 11
     1110111111111122111222211122
1300 data4644654557555456525473534363 75
     7373533555153523434363636565454 525
1310 data3757575555533133334325131313 313
     151535363434323252525454342444
1319 :
1320 data"Davis-sprong"
1330 data53,32,31,2211122221112211110 11
     1111111111111122111222211122
1340 data5153545256547555456531515153 53
     5533531333254545434323151313333735
     3252
1350 data5254545656367375755563656545 45
     2525232343573737353533 3353
1359 :
1360 data"Hoek[SPACE]naar[SPACE]hoek"
1370 data57,36,35,2201122211111211111 11
     11111111111111211111222 11122
1380 data5131434162424143444246442646 47
     4532523432543456543133343222425232
     545213
1390 data3314347353745475553436323437 35
     3436545652545755545666461535353745
     473757
1399 :
1400 data"Achtpuntige[SPACE]ster"
1410 data44,29,28,2201022211111201111 10
     1111111011111021111122201022
1420 data5557365657555456745444465636 26
     4647452426452526246664644434541434
1430 data6264644452543335543435333353 41
     435333234322424244
1439 :
1440 data"Vijfpuntige[SPACE]ster"
1450 data44,24,23,2201022211122011111 0
     1111111011111022111222200022
1460 data5557365644465755254546446545 23
     2543234543533333353515151313333331
     3151
1470 data51535355745455536343 4244
```

```
1479 :
1480 data"Vijf[SPACE]kruisen"
1490 data44,21,20,22010222011102010101O
     111111010101020111022201022
1500 data6466446474544262626464444444241
     4324222224242444424l434
1510 data4626262424444446474566464644
1519 :
1520 data"Dubbel[SPACE]kruis"
1530 data44,21,20,2201022210101200111OO
     111111100111002101012220l022
1540 data5456563626463555474545657454 66
     64436354746264
1550 data74543335543435334143432314342 2
     242444
1559 :
1560 data"Vierkant"
1570 data44,24,23,22010222011102011111O
     1110111011111020111022201022
1580 data537333534644654545434363353333
     31315151535355555757373 7353515
1590 data151313337353745454525232323424
     44
```

** EINDE LISTING solitair **

Checksum solitair

REGEL 1	16
REGEL 2	41
REGEL 3	36
REGEL 4	143
REGEL 10	196
REGEL 20	88
REGEL 30	78
REGEL 40	215
REGEL 50	154
REGEL 60	177
REGEL 70	156
REGEL 80	42
REGEL 90	150
REGEL 100	121
REGEL 110	249
REGEL 120	217
REGEL 130	29
REGEL 140	142
REGEL 150	238
REGEL 160	226
REGEL 170	12
REGEL 180	71
REGEL 190	100
REGEL 200	179
REGEL 210	106
REGEL 220	230
REGEL 230	7
REGEL 240	253
REGEL 250	2
REGEL 260	38
REGEL 270	76
REGEL 280	7
REGEL 290	147
REGEL 300	149
REGEL 310	102
REGEL 320	107
REGEL 330	178
REGEL 340	93
REGEL 350	28
REGEL 360	43
REGEL 370	134
REGEL 380	83
REGEL 390	196
REGEL 400	190
REGEL 410	151
REGEL 420	30
REGEL 430	88
REGEL 440	40
REGEL 450	61
REGEL 460	218
REGEL 470	68
REGEL 480	67
REGEL 490	34
REGEL 500	233
REGEL 510	188
REGEL 520	33
REGEL 530	107
REGEL 540	142
REGEL 550	206
REGEL 560	74

REGEL 570	19
REGEL 580	4
REGEL 590	229
REGEL 600	179
REGEL 610	12
REGEL 620	24
REGEL 630	102
REGEL 640	201
REGEL 650	222
REGEL 660	232
REGEL 670	157
REGEL 680	100
REGEL 690	131
REGEL 700	164
REGEL 710	231
REGEL 720	130
REGEL 730	34
REGEL 740	114
REGEL 750	84
REGEL 760	142
REGEL 770	78
REGEL 780	110
REGEL 790	149
REGEL 800	73
REGEL 810	194
REGEL 820	12
REGEL 830	230
REGEL 840	40
REGEL 850	203
REGEL 860	246
REGEL 870	128
REGEL 880	213
REGEL 890	214
REGEL 900	155
REGEL 910	145
REGEL 920	236
REGEL 930	51
REGEL 940	228
REGEL 950	122
REGEL 960	72
REGEL 970	45
REGEL 980	131
REGEL 990	122
REGEL 1000	149
REGEL 1010	250
REGEL 1020	30
REGEL 1028	58
REGEL 1029	58
REGEL 1030	125
REGEL 1040	40
REGEL 1050	179
REGEL 1059	58
REGEL 1060	154
REGEL 1070	49
REGEL 1080	55
REGEL 1089	58
REGEL 1090	61
REGEL 1100	133
REGEL 1110	179
REGEL 1119	58

REGEL 1120	62
REGEL 1130	141
REGEL 1140	227
REGEL 1149	58
REGEL 1150	242
REGEL 1160	57
REGEL 1170	55
REGEL 1179	58
REGEL 1180	222
REGEL 1190	144
REGEL 1200	29
REGEL 1209	58
REGEL 1210	185
REGEL 1220	148
REGEL 1230	231
REGEL 1239	58
REGEL 1240	120
REGEL 1250	151
REGEL 1260	171
REGEL 1270	87
REGEL 1279	58
REGEL 1280	158
REGEL 1290	168
REGEL 1300	161
REGEL 1310	153
REGEL 1319	58
REGEL 1320	196
REGEL 1330	168
REGEL 1340	63
REGEL 1350	251
REGEL 1359	58
REGEL 1360	183
REGEL 1370	176
REGEL 1380	149
REGEL 1390	221
REGEL 1399	58
REGEL 1400	193
REGEL 1410	169
REGEL 1420	167
REGEL 1430	47
REGEL 1439	58
REGEL 1440	208
REGEL 1450	162
REGEL 1460	69
REGEL 1470	103
REGEL 1479	58
REGEL 1480	151
REGEL 1490	145
REGEL 1500	7
REGEL 1510	63
REGEL 1519	58
REGEL 1520	131
REGEL 1530	145
REGEL 1540	151
REGEL 1550	191
REGEL 1559	58
REGEL 1560	171
REGEL 1570	154
REGEL 1580	183
REGEL 1590	24

sprong

Bij dit programma van F. Reijsbergen draait alles om mannetjes en vrouwtjes. De bedoeling is om door middel van schuiven en springen naast elk mannetje een vrouwtje te zetten. En dit alles in zo min mogelijk beurten. Denk erom, het lijkt makkelijker dan het is.

```
1    rem spring-sprong / cbm 64
2    rem door fons reijsbergen
3    rem
4    rem
10   gosub410:gosub250:gosub360
20   gosub170:gosub110:ifw=0then20
30   rem********win routine
40   printt$"Geweldig![SPACE]wat[SPACE]
     zijn[SPACE]ze[SPACE]blij,[SPACE]di
     e[SPACE]figuren[SPACE]hieronder."
50   print"[CRSR-DOWN]Het[SPACE]is[SPAC
     E]geluk[SPACE]in"t"sprongen."
60   printp$"[7xCRSR-DOWN]Nog[SPACE]een
     [SPACE]keertje[2xSPACE](j/n)"
70   geti$:ifi$="j"thenrun
80   ifi$="n"thenend
90   goto70
100  rem*******kontrole + sprong
110  w=0:fori=1to2:ifp(p+i)=3thenp(p+i)
     =p(p):p(p)=3:i=3:goto130
120  ifp(p-i)=3thenp(p-i)=p(p):p(p)=3:i
     =3
130  next:ifi<>4thenreturn
140  w=1:t=t+1:gosub220:fori=2to17:ifp(
     i)<>k(i)theni=17:w=0
150  next:return
160  rem*******input vis zwemt
170  geti$:p1=p:ifi$="[SPACE]"thenretur
     n
180  ifi$="z"andp>2thenp=p-1
190  ifi$="m"andp<16thenp=p+1
200  printp$"[4xCRSR-DOWN]"tab(41+p1*2)
     w$p$"[4xCRSR-DOWN]"tab(41+p*2)v$:g
     oto170
210  rem*******print poppen/score
220  printp$tab(5);:fori=2to17:printp$(
     p(i));:next
230  printt$tab(17)t:return
240  rem*******variable
250  dimp(18),k(35)
260  fori=2to8:p(i)=1:p(i+8)=2:next:p(9
     )=3:fori=2to16:readk(i):next:p=9:t
     =0
270  p$(1)="[CTRL-4][CTRL-9]AB[CRSR-DOW
     N][2xCRSR-LEFT]CD[CRSR-DOWN][2xCRS
     R-LEFT]EF[2xCRSR-UP][CTRL-0][CTRL-
     2]"
280  p$(2)="[CTRL-8][CTRL-9]GH[CRSR-DOW
     N][2xCRSR-LEFT]IJ[CRSR-DOWN][2xCRS
     R-LEFT]KL[2xCRSR-UP][CTRL-0][CTRL-
     2]"
290  p$(3)="[2xSPACE][CRSR-DOWN][2xCRSR
     -LEFT][2xSPACE][CRSR-DOWN][2xCRSR-
     LEFT][2xSPACE][2xCRSR-UP]":v$="[CT
     RL 2][CTRL-9]Q[CTRL-0]":w$="[COM-7
     ][CTRL-9]P[CTRL-0][CTRL-2]"
300  t$="[HOME][7xCRSR-DOWN]":p$=t$+"[9
     xCRSR-DOWN]"
310  return
320  rem*******print titel
330  print"[SHIFT-CLR][CTRL-2][7xSPACE]
     S[SPACE]P[SPACE]R[SPACE]I[SPACE]N[
     SPACE]G[SPACE]-[SPACE]S[SPACE]P[SP
     ACE]R[SPACE]O[SPACE]N[SPACE]G"
340  print"[CTRL-6][6xSPACE][27xSHIFT-*
     ]":return
350  rem*******print speelscherm
360  gosub330:printt$"[COM-6]Aantal[SPA
     CE]sprongen[SPACE]:"
370  printp$"[3xCRSR-DOWN][COM-7][CTRL-
     9]OOOOO[COM-1]";
380  fori=1to15:print"MN";:next:print"[
     COM-7]OOOOO";:fori=1to80:print"P";
     :next
390  printp$"[4xCRSR-DOWN]"tab(41+p*2)v
     $:gosub220:return
400  rem*******uitleg
410  poke53272,22:poke53280,0:poke53281
     ,0:poke657,128:gosub330
420  print"[COM-6][CRSR-DOWN]De[SPACE]b
     edoeling[SPACE]is[2xSPACE]dat[SPAC
     E]de[SPACE]7[SPACE]mannetjes[SPACE
     ]elk[SPACE]een[SPACE]vrouwtje[SPAC
     E]krijgen."
430  print"Natuurlijk[SPACE]willen[SPAC
     E]de[SPACE]mannetjes[SPACE]en[SPAC
     E]vrouw-";
440  print"tjes[SPACE]zo[SPACE]snel[SPA
     CE]mogelijk[SPACE]een[SPACE]partne
     r."
450  print"[CRSR-DOWN]Om[SPACE]dat[SPAC
     E]te[SPACE]doen[2xSPACE]kan[SPACE]
     de[SPACE]man[SPACE]of[SPACE]vrouw[
     SPACE]die"
460  print"naast[SPACE]een[SPACE]lege[S
     PACE]steen[SPACE]staat[SPACE]een[S
     PACE]plaatsje[SPACE]opschuiven."
470  print"Ook[SPACE]kan[SPACE]een[SPAC
     E]man[SPACE]of[SPACE]vrouw[SPACE]o
     ver[SPACE]een[SPACE]ander"
480  print"heen[SPACE]springen[SPACE]al
     s[SPACE]er[SPACE]naast[SPACE]die[S
     PACE]ander[SPACE]'n[SPACE]steen[SP
     ACE]vrij[SPACE]is."
490  print"[CRSR-DOWN]Het[SPACE]visje[S
     PACE]wijst[SPACE]aan[2xSPACE]welke
     [SPACE]man[SPACE]of[SPACE]vrouw"
500  print"de[SPACE]sprong[SPACE]gaat[S
     PACE]maken."
510  print"het[SPACE]visje[SPACE]is[SPA
     CE]te[SPACE]besturen[SPACE]met[SPA
     CE]de[SPACE]toetsen[2xSPACE][CTRL-
     6]Z[COM-6][SPACE]en[SPACE][CTRL-6]
     M[COM-6],[SPACE]met[SPACE]een[SPAC
     E]";
520  print"druk[2xSPACE]op[SPACE]de[SPA
     CE][CTRL-6]SPATIEBALK[COM-6][SPACE
     ]wordt[SPACE]de[SPACE]sprong[SPACE
     ]een[SPACE]feit."
530  print"[CRSR-DOWN][SPACE]De[SPACE]l
     inker[SPACE]steen[SPACE]is[SPACE]o
     p[SPACE]het[SPACE]einde[SPACE]leeg
     ."
540  print"De[SPACE]volgorde[SPACE]moet
     [SPACE]zijn[SPACE]man-vrouw-man-et
     c."
550  gosub590:print"[CTRL-2][10xSPACE]D
```

```
    RUK[SPACE]OP[SPACE]EEN[SPACE]TOETS
    [SPACE]!!!";:poke198,0
560 geti$:ifi$=""then560
570 :print"[SHIFT-CLR]":poke53272,30:r
    eturn
580 rem*******verpl.progr.mc/lees data
590 fori=49152to49204:reada:pokei,a:ne
    xt:sys49152
600 fori=15880to16015:reada:pokei,a:ne
    xt
610 return
620 rem-------verpl data
630 data169,0,141,14,220,169,51,133,1,
    169,0,133,251,133,253,169,48,133
640 data252,169,208,133,254,160,0,177,
    253,145,251,200,208,249,230,254
650 data230,252,169,80,197,252,208,239
    ,169,55,133,1,169,1,141,14,220,96,
    234
660 rem-------kar.data
670 data0,0,0,3,7,7,7,3,0,0,0,128,192,
    192,192,128
680 data1,31,31,23,19,17,23,23,0,240,2
    40,208,144,16,208,208
690 data51,50,2,2,2,6,14,14,152,152,12
    8,128,128,192,224,224
700 data0,0,0,3,7,7,15,11,0,0,0,128,19
    2,192,224,160
710 data1,15,15,11,9,11,15,31,0,224,22
    4,160,32,160,224,240
720 data7,7,2,2,2,2,6,14,192,192,128,1
    28,128,128,192,224
730 data0,31,15,63,63,127,120,0,0,240,
    252,252,254,255,102,0,0,0,0,0,0
740 data117,142,0,117,142,0,0,142,117,
    0,0,24,68,94,125,127,94,68,24
750 data3,1,2,1,2,1,2,1,2,1,2,1,2,1,2
760 :******klaar met het typewerk*****
    **
```

** EINDE LISTING sprong **

wormy

Ryan van Hattum is de maker van dit spel. Het is een heel oud gegeven, een slang moet sterretjes op eten en wordt daardoor steeds langer. Hierdoor wordt het steeds moeilijker om goed en snel te reageren. Echt een spelletje om niet bij te hoeven nadenken, ontspannend dus.

```
80  rem"[7xDEL][SPACE]wormy
90  rem"[7xDEL]
100 poke53280,0:poke53281,0:clr
110 poke808,234
120 print"[SHIFT-CLR][CTRL-2]een[SPACE
    ]moment..."
130 dimhi$(5),hi(5),s(1100)
140 rem  hi-score loader...
150 open2,8,2:close2
160 ifst=-128thengoto850
170 open1,8,0,"hi-score[2xSPACE]/wormy,u,r"
180 fora=1to5:input#1,hi$(a):nexta
190 fora=1to5:input#1,hi(a):nexta
200 close1
```

Checksum sprong

REGEL 1	211	REGEL 390	59
REGEL 2	41	REGEL 400	127
REGEL 3	36	REGEL 410	32
REGEL 4	143	REGEL 420	101
REGEL 10	224	REGEL 430	253
REGEL 20	133	REGEL 440	98
REGEL 30	243	REGEL 450	235
REGEL 40	1	REGEL 460	163
REGEL 50	44	REGEL 470	138
REGEL 60	38	REGEL 480	134
REGEL 70	177	REGEL 490	189
REGEL 80	99	REGEL 500	246
REGEL 90	240	REGEL 510	58
REGEL 100	39	REGEL 520	242
REGEL 110	129	REGEL 530	7
REGEL 120	252	REGEL 540	164
REGEL 130	93	REGEL 550	240
REGEL 140	109	REGEL 560	120
REGEL 150	74	REGEL 570	213
REGEL 160	206	REGEL 580	38
REGEL 170	40	REGEL 590	74
REGEL 180	255	REGEL 600	104
REGEL 190	40	REGEL 610	142
REGEL 200	36	REGEL 620	109
REGEL 210	191	REGEL 630	111
REGEL 220	182	REGEL 640	73
REGEL 230	97	REGEL 650	125
REGEL 240	251	REGEL 660	240
REGEL 250	192	REGEL 670	76
REGEL 260	106	REGEL 680	59
REGEL 270	129	REGEL 690	69
REGEL 280	165	REGEL 700	162
REGEL 290	136	REGEL 710	252
REGEL 300	87	REGEL 720	192
REGEL 310	142	REGEL 730	254
REGEL 320	196	REGEL 740	207
REGEL 330	78	REGEL 750	211
REGEL 340	3	REGEL 760	16
REGEL 350	125		
REGEL 360	34		
REGEL 370	247		
REGEL 380	40		

```
210 open15,8,15:input#15,e:close15
220 ife<>0thengoto850
230 goto960
240 sc=0:poke646,12
250 print"[SHIFT-CLR][CTRL-4]"tab(14)"
    ***[SPACE]wormy[SPACE]***[CTRL-8]"
    :a=rnd(0)
260 poke53280,0:poke53281,0:fori=0to19
    :poke1083-i,160:poke1084+i,160:nexti
270 fori=0to21:poke1104+i*40,160:poke1
    143+i*40,160:nexti
280 fori=0to19:poke1984+i,160:poke2023
    -i,160:nexti
290 wh=292:wt=wh:d=1:pokewh+1024,81:s(
    wh)=d:gosub540:gosub560
300 qy=peek(56320)and16
310 ifqy=0thengosub810
320 q=peek(56320)and15
330 ifq=11thend=-1
340 ifq=7thend=1
350 ifq=14thend=-40
360 ifq=13thend=40
370 ifq=10thend=-41
380 ifq=6thend=-39
390 ifq=9thend=79
400 ifq=5thend=81
410 ifd=-dlthendl=d
420 s(wh)=d:wh=wh+d
430 ifpeek(wh+1024)=90thensc=sc+10:gos
    ub560:gosub540:l=5:goto450:gosub560
440 ifpeek(wh+1024)<>32then490
450 pokewh+1024,81:iflthenl=l-1:goto480
```

```
460 pokewt+1024,32:wt=wt+s(wt)
470 gosub560
480 goto300
490 p=peek(wh+1024):pokewh+1024,86:pok
    e55296+wh,1
500 gosub620
510 q=peek(56320)and16
520 ifq=0then720
530 goto510
540 r=int(rnd(1)*960)+1064:ifpeek(r)<>
    32then540
550 poker,90:return
560 printchr$(19)"score:[SPACE]"sc:return
570 goto1130
580 poke53280,15:fora=0to100:nexta
590 qy=peek(56320)and16
600 ifqy<>0then590
610 poke53280,0:return
620 print"[HOME][12xCRSR-DOWN][16xCRSR
    -RIGHT][CTRL-1]game[SPACE]over"
630 print"[HOME][12xCRSR-DOWN][16xCRSR
    -RIGHT][COM-4]game[SPACE]over"
640 fora=0to50:nexta
650 print"[HOME][12xCRSR-DOWN][16xCRSR
    -RIGHT][COM-5]game[SPACE]over"
660 fora=0to50:nexta
670 print"[HOME][12xCRSR-DOWN][16xCRSR
    -RIGHT][COM-8]game[SPACE]over"
680 fora=0to50:nexta
690 print"[HOME][12xCRSR-DOWN][16xCRSR
    -RIGHT][CTRL-2]game[SPACE]over"
700 fora=0to50:nexta
710 return
720 print"[HOME][12xCRSR-DOWN][16xCRSR
    -RIGHT][CTRL-2]game[SPACE]over"
730 fora=0to50:nexta
740 print"[HOME][12xCRSR-DOWN][16xCRSR
    -RIGHT][COM-5]game[SPACE]over"
750 fora=0to50:nexta
760 print"[HOME][12xCRSR-DOWN][16xCRSR
    -RIGHT][COM-4]game[SPACE]over"
770 fora=0to50:nexta
780 print"[HOME][12xCRSR-DOWN][16xCRSR
    -RIGHT][CTRL-1]game[SPACE]over"
790 fora=0to50:nexta
800 goto570
810 poke53280,15:fora=0to250:nexta
820 qy=peek(56320)and16
830 ifqy<>0then820
840 poke53280,0:return
850 hi$(1)=".....wormy!....."
860 hi(1)=250
870 hi$(2)="...geschreven..."
880 hi(2)=200
890 hi$(3)="......voor......"
900 hi(3)=150
910 hi$(4)="..c=info..door.."
920 hi(4)=100
930 hi$(5)=".tonnie[SPACE]en[SPACE]ryan."
940 hi(5)=50
950 goto230
960 rem  beginbeeld
970 print"[SHIFT-CLR]"chr$(142)chr$(8)
980 print"[2xCRSR-DOWN][COM-6][14xSPAC
    E]***[SPACE]wormy[SPACE]***"
990 print"[COM-8][2xCRSR-DOWN][2xSPACE
    ]geschreven[SPACE]door:[SPACE]ryan
    [SPACE]van[SPACE]hattum[SPACE]en"
1000 print"[14xSPACE]tonnie[SPACE]faber!"
1010 print"[2xCRSR-DOWN][COM-6][2xSPACE
     ]om[SPACE]te[SPACE]beginnen[SPACE]
     druk[SPACE]op[SPACE]de[SPACE]vuurknop.."
1020 fora=1to500
1030 b=peek(56320)and16
1040 ifb=0thengoto240
1050 nexta
1060 print"[SHIFT-CLR][3xCRSR-DOWN][COM
     6][11xSPACE]***[SPACE]hi-scores[S
     PACE]***"
1070 print"[2xCRSR-DOWN][COM-8]"
1080 fora=1to5:print"[4xSPACE]"a"[SPACE
     ]>>[SPACE]"hi$(a)"[SPACE]<<[SPACE]
     "hi(a):nexta
1090 fora=1to500
1100 b=peek(56320)and16
1110 ifb=0then240
1120 nexta:goto960
1130 rem  hi-score gehaald ja/nee?
1140 ifsc<=hi(5)orsc=hi(4)orsc=hi(3)ors
     c=hi(2)orsc=hi(1)thengoto230
1150 print"[SHIFT-CLR][CRSR-DOWN][CTRL-
     8][2xSPACE]u[SPACE]heeft[SPACE]een
     [SPACE]hi-score[SPACE]plaats[SPACE
     ]behaald!"
1160 print"[2xCRSR-DOWN][COM-3]type[SPA
     CE]een[SPACE]naam[SPACE]>>[SPACE]"
     ;:poke19,1:inputa$:poke19,0
1170 iflen(a$)>16thena$=left$(a$,16):go
     to1200
1180 a=(16-len(a$)):forb=1toa
1190 a$=a$+"[SPACE]":nextb
1200 ifsc>hi(5)andsc<hi(4)thenhi(5)=sc:
     hi$(5)=a$:goto1330
1210 r=0
1220 ifsc>hi(4)andsc<hi(3)thenhi$(5)=hi
     $(4):hi(5)=hi(4):hi$(4)=a$:hi(4)=s
     c:r=1
1230 ifr=0thengoto1250
1240 goto1330
1250 ifsc>hi(3)andsc<hi(2)thenhi$(5)=hi
     $(4):hi(5)=hi(4):hi$(4)=hi$(3):r=1
1260 ifr=0thengoto1280
1270 hi(4)=hi(3):hi$(3)=a$:hi(3)=sc:got
     o1330
1280 ifsc>hi(2)andsc<hi(1)thenhi$(5)=hi
     $(4):hi(5)=hi(4):hi$(4)=hi$(3):r=1
1290 ifr=0thengoto1310
1300 hi(4)=hi(3):hi$(3)=hi$(2):hi(3)=hi
     (2):hi$(2)=a$:hi(2)=sc:goto1330
1310 hi$(5)=hi$(4):hi(5)=hi(4):hi$(4)=h
     i$(3):hi(4)=hi(3):hi$(3)=hi$(2)
1320 hi(3)=hi(2):hi$(2)=hi$(1):hi(2)=hi
     (1):hi$(1)=a$:hi(1)=sc
1330 printchr$(13)"[2xCRSR-DOWN][COM-5]
     save[SPACE]hi-score?[SPACE](j/n)"
1340 poke198,0
1350 geta$:ifa$<>"j"anda$<>"n"then1350
1360 ifa$="n"thengoto1060
1370 open2,8,2:close2
1380 ifst=-128then1460
1390 open1,8,1,"`0:hi-score[2xSPACE]/wo
     rmy,u,w"
1400 fora=1to5:print#1,hi$(a):nexta
1410 fora=1to5:print#1,hi(a):nexta
1420 close1
1430 open15,8,15:input#15,e:close15
```

```
1440 ife<>0thengoto1460
1450 goto1060
1460 print"[2xCRSR-DOWN][CTRL-2]disk[SPACE]error![SPACE](druk[SPACE]op[SPACE]de[SPACE]vuurknop)"
1470 b=peek(56320)and16:ifb<>0then1470
1480 goto1060
```

** EINDE LISTING wormy **

Checksum wormy

REGEL 80	219	REGEL 810	65
REGEL 90	61	REGEL 820	133
REGEL 100	251	REGEL 830	10
REGEL 110	252	REGEL 840	189
REGEL 120	167	REGEL 850	184
REGEL 130	150	REGEL 860	92
REGEL 140	10	REGEL 870	44
REGEL 150	159	REGEL 880	88
REGEL 160	247	REGEL 890	157
REGEL 170	6	REGEL 900	93
REGEL 180	221	REGEL 910	36
REGEL 190	185	REGEL 920	89
REGEL 200	209	REGEL 930	39
REGEL 210	208	REGEL 940	46
REGEL 220	49	REGEL 950	30
REGEL 230	40	REGEL 960	80
REGEL 240	120	REGEL 970	111
REGEL 250	37	REGEL 980	50
REGEL 260	198	REGEL 990	119
REGEL 270	117	REGEL 1000	43
REGEL 280	107	REGEL 1010	87
REGEL 290	48	REGEL 1020	222
REGEL 300	133	REGEL 1030	29
REGEL 310	228	REGEL 1040	117
REGEL 320	43	REGEL 1050	195
REGEL 330	105	REGEL 1060	197
REGEL 340	147	REGEL 1070	154
REGEL 350	159	REGEL 1080	185
REGEL 360	243	REGEL 1090	222
REGEL 370	156	REGEL 1100	29
REGEL 380	120	REGEL 1110	236
REGEL 390	212	REGEL 1120	37
REGEL 400	201	REGEL 1130	128
REGEL 410	179	REGEL 1140	148
REGEL 420	81	REGEL 1150	185
REGEL 430	70	REGEL 1160	214
REGEL 440	187	REGEL 1170	113
REGEL 450	121	REGEL 1180	148
REGEL 460	127	REGEL 1190	104
REGEL 470	40	REGEL 1200	19
REGEL 480	28	REGEL 1210	52
REGEL 490	34	REGEL 1220	110
REGEL 500	37	REGEL 1230	183
REGEL 510	44	REGEL 1240	80
REGEL 520	254	REGEL 1250	168
REGEL 530	31	REGEL 1260	186
REGEL 540	58	REGEL 1270	136
REGEL 550	70	REGEL 1280	166
REGEL 560	115	REGEL 1290	180
REGEL 570	78	REGEL 1300	248
REGEL 580	59	REGEL 1310	19
REGEL 590	133	REGEL 1320	102
REGEL 600	14	REGEL 1330	211
REGEL 610	189	REGEL 1340	149
REGEL 620	114	REGEL 1350	156
REGEL 630	121	REGEL 1360	43
REGEL 640	170	REGEL 1370	159
REGEL 650	122	REGEL 1380	156
REGEL 660	170	REGEL 1390	182
REGEL 670	125	REGEL 1400	241
REGEL 680	170	REGEL 1410	205
REGEL 690	231	REGEL 1420	209
REGEL 700	170	REGEL 1430	208
REGEL 710	142	REGEL 1440	95
REGEL 720	231	REGEL 1450	80
REGEL 730	170	REGEL 1460	19
REGEL 740	122	REGEL 1470	43
REGEL 750	170	REGEL 1480	8
REGEL 760	121		
REGEL 770	170		
REGEL 780	114		
REGEL 790	170		
REGEL 800	37		

beveilig

Er wordt ons vaak gevraagd of het mogelijk is basic programma's te beveiligen. Echt beveiligen is niet mogelijk, een goede programmeur kan dit altijd omzeilen. Om een listing wat moeilijker leesbaar te maken zijn er verschillende mogelijkheden. Denk maar eens aan het manipuleren met kleuren, voorgrond gelijk aan de achtergrond kleur. Verschillende poke opdrachten werken ook, maar eenvoudiger gaat het met het volgende programma van Marco Paauwe.

```
10   fori=49730to49886:reada:b=b+a:pokei,a:next
20   ifb<>22263thenprint"fout[SPACE]in[SPACE]data":end
30   print"[SHIFT-CLR]laadt[SPACE]nu[SPACE]het[SPACE]basic[SPACE]programma[SPACE]in[SPACE]dat[SPACE]"
40   print"dat[SPACE]beveiligt[SPACE]moet[SPACE]worden[SPACE]en[SPACE]beveilig[SPACE]"
50   print"met[SPACE]sys49730"
60   print"[CRSR-DOWN]waarschuwing[SPACE]eenmaal[SPACE]beveiligd[SPACE]kun[SPACE]je"
70   print"het[SPACE]programma[SPACE]niet[SPACE]terug[SPACE]veranderen"
80   print"na[SPACE]het[SPACE]beveiligen[SPACE]wegsaven"
1000 data160,0,165,45,133,171,165,46,133,172
1001 data165,55,133,169,165,56,133,170,198,171
1002 data165,171,201,255,208,2,198,172,198,169
1003 data165,169,201,255,208,2,198,170,177,171
1004 data145,169,165,171,197,43,165,172,229,44
1005 data176,222,160,1,177,169,200,17,169,240
1006 data67,160,0,177,169,145,171,200,192,5
1007 data208,247,165,171,24,105,5,133,171,144
1008 data2,230,172,160,0,152,145,171,200,169
1009 data58,192,5,208,247,177,169,240,6,145
1010 data171,200,76,161,194,152,24,101,169,133
1011 data169,144,2,230,170,152,24,101,171,133
1012 data171,144,185,230,172,76,118,194,160,0
1013 data152,145,171,200,192,3,208,249,152,24
1014 data101,171,133,45,165,172,105,0,133,46
1015 data76,171,225,96,96,165,142
```

** EINDE LISTING beveilig **

Checksum beveilig

REGEL 10	218	REGEL 1006	24
REGEL 20	93	REGEL 1007	117
REGEL 30	210	REGEL 1008	52
REGEL 40	47	REGEL 1009	41
REGEL 50	201	REGEL 1010	164
REGEL 60	124	REGEL 1011	103
REGEL 70	5	REGEL 1012	122
REGEL 80	135	REGEL 1013	114
REGEL 1000	66	REGEL 1014	55
REGEL 1001	189	REGEL 1015	5
REGEL 1002	189		
REGEL 1003	184		
REGEL 1004	193		
REGEL 1005	122		

kleur

Van Joris van den Heuvel en Hanno Krijgsman kregen wij het programma Schermkleur. Met dit programma kan men de border- de scherm en de cursorkleur blijvend veranderen. Dat wil zeggen, elke keer dat **READY** verschijnt, worden de kleuren terug gezet, die u heeft ingevoerd. Dit is zeer bruikbaar voor de mensen die 's avonds lang voor het scherm zitten. Als u nu rustige kleuren ingeeft, heeft dat veel voordelen voor de ogen. Een tip: zeer rustige stabiele kleuren zijn: border: donkergrijs (11), scherm : donkergrijs (11), cursor: Licht grijs (15).

```
100  rem scherm-kleur / c=64
110  rem
120  rem
130  rem Joris van den Heuvel
140  rem
150  :
160  fort=0to45:reada:b=b+a:poke53200+t
     ,a:next
170  ifb<>5480thenprint"[SHIFT-CLR][CRS
     R-DOWN][CRSR-RIGHT]fout[SPACE]in[S
     PACE]data[SPACE]!":stop
180  input"[SHIFT-CLR][CRSR-DOWN][CRSR-
     RIGHT]-border-kleur[SPACE]";b
190  input"[CRSR-DOWN][CRSR-RIGHT]-sche
     rm-kleur[SPACE]";s
200  input"[CRSR-DOWN][CRSR-RIGHT]-curs
     or-kleur[SPACE]";c
210  poke53230,b:poke53235,s:poke53240,
     c
220  print"[7xCRSR-DOWN]run/stop[SPACE]
     +[SPACE]restore[2xSPACE]zet[SPACE]
     de[SPACE]kleuren[SPACE]niet[SPACE]
     meer[SPACE]terug!"
230  print"[CRSR-DOWN]na[SPACE]reset[SP
     ACE]weer[SPACE]inschakelen[SPACE]m
     et[SPACE]sys53210"
240  gett$:ift$<>""thensys53200
250  goto240
260  data 173,2,3,201,237,240,19,141
270  data 253,207,173,3,3,141,254,207
280  data 169,237,141,2,3,169,207,141
290  data 3,3,76,102,254,169,14,141
300  data 32,208,169,6,141,33,208,169
310  data 14,141,134,2,76,59,223
```

** EINDE LISTING kleur **

Checksum kleur

REGEL 100	26	REGEL 240	61
REGEL 110	155	REGEL 250	31
REGEL 120	227	REGEL 260	124
REGEL 130	227	REGEL 270	181
REGEL 140	217	REGEL 280	189
REGEL 150	58	REGEL 290	83
REGEL 160	19	REGEL 300	194
REGEL 170	33	REGEL 310	194
REGEL 180	162		
REGEL 190	36		
REGEL 200	48		
REGEL 210	146		
REGEL 220	160		
REGEL 230	56		

PRINT OUT C-128 met o.a. Hannoi

lucifer

Spelen waarbij moet worden nagedacht blijven het goed doen. Bij dit spel is diegene die de laatste lucifer pakt de verliezer. Er mogen er per beurt 1,2 of 3 worden opgepakt. Dus veel spelen en blijven opletten om het geheim van winst te ontdekken.

```
100 rem *** lucifer-spel
110 poke53280,0:poke53281,0:x=21:print
    chr$(142)chr$(11)"[SHIFT-CLR][CTRL
     6]wie[SPACE]begint[SPACE]er[SPACE]";
120 pc=int(rnd(1)*3)+1:print"(computer
    /speler)[SPACE]??":l$="[CTRL-3]Q[C
    RSR-DOWN][CRSR-LEFT][CTRL-8][SHIFT
     -][CRSR-DOWN][CRSR-LEFT][SHIFT--]
    [CRSR-DOWN][CRSR-LEFT][SHIFT--][CR
    SR-DOWN][CRSR-LEFT][SHIFT--][4xCRSR-
    UP]"
130 gosub500:ifa$="c"then210
140 ifa$<>"s"then130
150 print"[SHIFT-CLR]";:fort=1tox:prin
    tl$;:nextt:print"[6xCRSR-DOWN]":pr
    int"[CTRL-6]aantal[SPACE]lucifers:";x
160 ifx=1thenprint"[CRSR-DOWN]jij[SPAC
    E]moet[SPACE]er[SPACE]1[SPACE]neme
    n,":print"ik[SPACE]heb[SPACE]gewon
    nen[SPACE]!!":goto270
170 print"[CRSR-DOWN]hoeveel[SPACE]nee
    m[SPACE]je[SPACE]??[SPACE]";
180 gosub500:pc=val(a$):ifpc<1orpc>3or
    pc=>xthen180
190 pc=int(pc):printa$:x=x-pc:print"[C
    RSR-DOWN]nu[SPACE]nog[SPACE]over:"
    ;x:pc=int(rnd(ti)*3)+1
200 print"[HOME][24xCRSR-DOWN]druk[SPA
    CE]op[SPACE]een[SPACE]toets.[HOME]
    ":gosub500
210 print"[SHIFT-CLR]";:fort=1tox:prin
    tl$;:nextt:print"[6xCRSR-DOWN]":pr
    int"[CTRL-6]aantal[SPACE]lucifers:";x
220 ifx=1thenprint"[CRSR-DOWN]ik[SPACE
    ]moet[SPACE]er[SPACE]1[SPACE]nemen
    ,":print"jij[SPACE]hebt[SPACE]gewo
    nnen[SPACE]!!":goto270
230 ifx<18thenfort=0to2:if(x+t)/4=int(
    (x+t)/4)thenpc=3-t:goto250
240 ifx<18thennextt
250 print"[CRSR-DOWN]ik[SPACE]neem[SPA
    CE]er";pc:x=x-pc:print"[CRSR-DOWN]
    nu[SPACE]nog[SPACE]over:";x
260 print"[HOME][24xCRSR-DOWN]druk[SPA
    CE]op[SPACE]een[SPACE]toets.[HOME]
    ":gosub500:goto150
270 print"[HOME][24xCRSR-DOWN]druk[SPA
    CE]op[SPACE]een[SPACE]toets.[HOME]
    ":gosub500
280 print"[SHIFT-CLR]nog[SPACE]een[SPA
    CE]keer[SPACE](ja/nee)[SPACE]??"
290 gosub500:ifa$="j"thenrun
300 ifa$<>"n"then290
310 print"[SHIFT-CLR]":end
320 :
500 rem *** druk op een toets
510 geta$:ifa$<>""thenreturn
520 goto510
```

** EINDE LISTING lucifer **

Checksum Lucifer

REGEL 100	120	REGEL 250	158
REGEL 110	45	REGEL 260	186
REGEL 120	44	REGEL 270	97
REGEL 130	191	REGEL 280	180
REGEL 140	38	REGEL 290	189
REGEL 150	180	REGEL 300	40
REGEL 160	77	REGEL 310	42
REGEL 170	99	REGEL 320	58
REGEL 180	189	REGEL 500	73
REGEL 190	144	REGEL 510	13
REGEL 200	97	REGEL 520	31
REGEL 210	180		
REGEL 220	161		
REGEL 230	56		
REGEL 240	124		

hannoi

De toren van Hannoi okk al zo een oud spel. Ditmaal bewerkt voor onze trouwe commodore 64. Het spel bestaat uit een stapel schijven in aflopende grootte. De bedoeling is om in zo min mogelijk beurten de toren over te stapelen, waarbij slechts een hulppositie mag worden gebruikt en de nieuwe schijf moet kleiner zijn dan de schijf welke er al ligt. Let erop want een (klein) foutje is zo gemaakt.

```
10  color0,12:color4,12:graphic0,1:win
    dow0,0,39,24:print"[SHIFT-CLR][COM
     6]s[SPACE]p[SPACE]e[SPACE]l[SPACE
    ]r[SPACE]e[SPACE]g[SPACE]e[SPACE]l
    [SPACE]s"
20  print"[19xCOM-T][CTRL-6]":print"he
    t[SPACE]is[SPACE]de[SPACE]bedoelin
    g[SPACE]om[SPACE]een[SPACE]aantal[
    SPACE](4-12)
30  print"ringen[SPACE]van[SPACE]paal[
    SPACE]1[SPACE]naar[SPACE]paal[SPAC
    E]3[SPACE]te[SPACE]brengen":print"
    op[SPACE]de[SPACE]volgende[SPACE]m
    anier:
40  print"[CRSR-DOWN][COM-6]*[CTRL-6][
    SPACE]er[SPACE]kan[SPACE]slechts[S
    PACE]een[SPACE]ring[SPACE]per[SPAC
    E]keer":print"[2xSPACE]verplaatst[
    SPACE]worden;
50  print"[COM-6]*[CTRL-6][SPACE]verpl
    aatsen[SPACE]kan[SPACE]alleen[SPAC
    E]naar[SPACE]een":print"[2xSPACE]a
    ndere[SPACE]paal[SPACE]waar[SPACE]
    geen[SPACE]";
60  print"of[SPACE]een[SPACE]kleinere"
    :print"[2xSPACE]ring[SPACE]als[SPA
    CE]bovenste[SPACE]zit.[CRSR-DOWN]"
70  print"veel[SPACE]plezier[SPACE]met
    [SPACE]het[SPACE]spel[SPACE]!!":pr
    int"[CRSR-DOWN]druk[SPACE]op[SPACE
    ]een[SPACE]toets.":getkeya$
80  print"[SHIFT-CLR][CTRL-6]hoogte[SP
    ACE](4-12)[SPACE]?[SPACE][COM-6][C
    OM `][CRSR-LEFT]";:do:getkeya$:ifa
    $<"1"ora$>"9"thenloop
90  ifa$="2"ora$="3"thenloop:elseprint
    a$"[COM-`][CRSR-LEFT]";:ifa$>"3"th
    enho=val(a$):goto110
100 getkeyb$:ho=val(a$+b$):ifho>12orho
    <10then100:elseprintb$;
```

```
110 print"[SPACE]":dima(3,ho+1),a$(ho)
    ,b(3):fort=1to3:a(t,ho+1)=ho+1:nex
    t:a$(0)="[CTRL-6][SHIFT--][COM-6]"
120 fort=1tohostep4:a$(t)="[CTRL-9]"+l
    eft$("[10xSPACE]",(t-1)/2+1)+"[CTR
    L 0]":nextt
130 fort=2tohostep4:a$(t)="[COM-M]"+a$
    (t-1)+"[CTRL-0][COM-G]":nextt:fort
    =3tohostep4
140 a$(t)="[CTRL-9][COM-K]"+a$(t-2)+"[
    CTRL-0][COM-K]":nextt:fort=4tohost
    ep4:a$(t)="[CTRL-9][COM-G]"+a$(t-3
    )+"[CTRL-9][COM-M][CTRL-0]"
150 nextt:b(1)=1:b(2)=ho+1:b(3)=ho+1:f
    ort=1toho:a(1,t)=t:nextt
160 print"[CRSR-DOWN][CTRL-6]hier[SPAC
    E]heb[SPACE]je[SPACE][COM-6]minste
    ns"2[KWADRAAT PIJL]ho-1"beurten[CT
     RL 6][SPACE]voor":print"nodig.[SPA
     CE]weet[SPACE]";
170 print"het[SPACE]zeker[SPACE](j/n)[
    SPACE]?[SPACE][COM-6][COM-'][CRSR-
    LEFT]";:do:getkeya$:loopuntila$="j
    "ora$="n":print"[SHIFT-CLR]"
180 ifa$="n"thenend:elsefort=1to3:prin
    t"[HOME]";:forr=0toho
190 printtab(t*13-6-(a(t,r)+3)/4)a$(a(
    t,r)):nextr,t
200 print"[CTRL-6][6xCOM-U][CTRL-9]1[C
    TRL 0][12xCOM-U][CTRL-9]2[CTRL-0][
    12xCOM-U][CTRL-9]3[CTRL-0][6xCOM-U
    ][CRSR-DOWN]"
210 br=br+1:char1,28,ho+3,chr$(27)+"pb
    eurt:[SPACE]":printusing"[COM-6]##
    ##";br
220 print"[CRSR-DOWN][COM-6]s[CTRL-6][
    SPACE]=[SPACE]stoppen":char1,0,ho+
    3,"[CTRL-6]van[SPACE][COM-6][COM-'
    ][CRSR-LEFT]":do:getkeya$:ifa$="s"
    then330
230 a=val(a$):ifa<1ora>3thenloop:elsei
    fb(a)>hothenloop:elsefort=1to3
240 ifa(a,b(a))<a(t,b(t))thenprinta$"[
    CTRL-6][SPACE]naar[SPACE][COM-6][C
    OM '][CRSR-LEFT]";:goto250:elsenex
    t:loop
250 do:getkeyb$:ifb$="s"then330:elseb=
    val(b$):ifb<1orb>3orb=athenloop
260 ifa(a,b(a))>a(b,b(b))thenloo'p:els
    eprintb$:char1,13*a-11,b(a),"[4xSP
    ACE][CTRL-6][SHIFT--][COM-6][4xSPA
    CE]"
270 char1,13*b-6-(a(a,b(a))+3)/4,b(b)-
    1,a$(a(a,b(a))):a(b,b(b)-1)=a(a,b(
    a))
280 a(a,b(a))=0:b(a)=b(a)+1:b(b)=b(b)-
    1:ifb(3)<>1then210
290 char1,0,ho+3,chr$(27)+"'[CTRL-6]he
    t[SPACE]is[SPACE]je[SPACE]gelukt[S
    PACE]in[COM-6]":printbr"[CTRL-6]be
    urten[SPACE]!!"
300 ifbr=2[KWADRAATPIJL]ho-1thenprint"
     het[SPACE]kan[SPACE]niet[SPACE]sne
     ller[SPACE]!!":goto320
310 print"het[SPACE]kon[SPACE]sneller,
    [SPACE]in"2[KWADRAAT PIJL]ho-1"beu
     rten[SPACE]!!"
320 print"[CRSR-DOWN]druk[SPACE]op[SPA
    CE]een[SPACE]toets.[COM-6]":getkey
    a$:graphic0,1:end
330 br=br-1:char1,0,ho+3,chr$(27)+"'[C
    TRL 6]":print"je[SPACE]hebt[SPACE]
    je[SPACE]taak[SPACE]niet[SPACE]vol
    bracht[SPACE]!!"
340 print"[CRSR-DOWN]laf[SPACE]hoor,[S
    PACE]om[SPACE]er[SPACE]na[COM-6]"b
    r"beurt";:ifbr>1orbr=0thenprint"en
    ";
350 print"[CTRL-6][SPACE]al":print"van
    af[SPACE]te[SPACE]zien[SPACE]!!":g
    oto320
```

** EINDE LISTING hannoi **

Checksum Hannoi

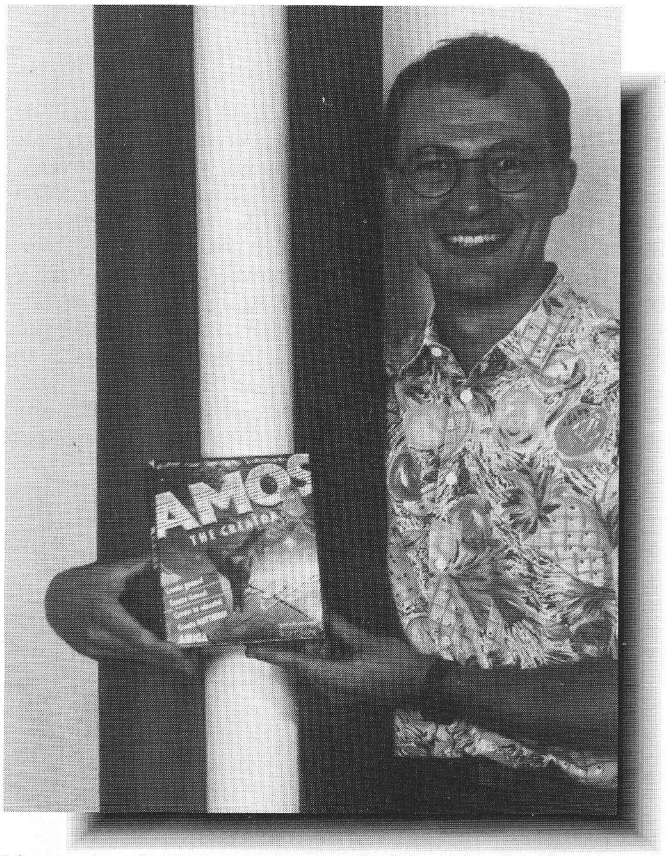
REGEL 10	112	REGEL 220	0
REGEL 20	210	REGEL 230	167
REGEL 30	72	REGEL 240	89
REGEL 40	113	REGEL 250	142
REGEL 50	241	REGEL 260	236
REGEL 60	92	REGEL 270	255
REGEL 70	44	REGEL 280	240
REGEL 80	67	REGEL 290	46
REGEL 90	200	REGEL 300	66
REGEL 100	90	REGEL 310	108
REGEL 110	124	REGEL 320	137
REGEL 120	26	REGEL 330	134
REGEL 130	131	REGEL 340	200
REGEL 140	62	REGEL 350	116
REGEL 150	190		
REGEL 160	210		
REGEL 170	211		
REGEL 180	136		
REGEL 190	112		
REGEL 200	110		
REGEL 210	227		

François Lionet

De man achter AMOS

Eindelijk was het dan zover. Het veel besproken AMOS verscheen bij de redactie als demo en als complete uitvoering, direct uit Engeland. Even gingen er stemmen op dat het wel een Amiga-versie van de succesvolle STOS basic taal voor de Atari ST zou gaan, maar niets was minder waar. Deze basic-taal was compleet herschreven: een nieuw programma.

De Franse schrijver van deze krachtige BASIC-programmeertaal, Francois Lionet, die ook verantwoordelijk was voor het kassucces STOS voor de AtariST, begint een levende legende te worden. Eerder dit jaar had het Newsflash magazine (03404-52987) een interview met hem.



AMOS - De persoon

Newsflash: Goed, vertel eerst eens iets over jezelf voordat we het over AMOS gaan hebben. Waar ben je geboren en waar woon je precies?
François: Ik woon in Yerres, aan kleine plaats ongeveer 25 kilometer van Parijs. Ik werd op 7 juni 1963 geboren in Maubeuge, een dorp dicht tegen de grens met Belgie. Verder heb ik twee broers waarvan de een tandarts is en de ander cardioloog.
Zelf ben ik getrouwd met Carine en onze hond heet Daisy. En je zult het niet geloven, ik haalde mijn dierenarts-diploma. Maar doordat ik het programmeren leuker vindt is hier verder nietsmeer van gekomen.
N: En hoe zit het met je eten?
F: Alleen het beste. Ik hou van veel dingen en ga graag naar HELE goede restaurants, zo'n een keer in de twee maanden. Ik vind dat het geld wel waard. Eigenlijk zou ik zo niet eens iets weten dat ik niet lekker vindt. Mischien zeeslakken of zo...
N: Hou je van muziek?
F: Alleen goede muziek... en vooral de laatste CD die ik heb gekocht. Ik heb die goede muziek echt nodig als ik programmeer. Daardoor werk ik gewoon beter. Ik bedoel dan niet als achtergrondmuziek, maar echt heel hard door een koptelefoon. Het is soms zo erg, dat ik wilde dat ik een lamp op m'n telefoon had, omdat ik dan het ding niet over hoor gaan. Een groep die je in Nederland niet hebt, en die ik echt slecht vindt, is "Les Muscles".
N: Waar ga je het liefste heen en waar het minste?
F: Ik ga het liefst naar de bergen, en een plek waar ik ronduiteen hekel aan heb zijn kleine steden.
N: Wat is voor jou de gebeurtenis van de jaren '80?
F: De 80'er jaren... Ik weet niet... Misschien net als iedereen de veranderingen in Oost Europa.
N: Waar hoop je in dit opzicht op?
F: Dat Gorby er in ieder geval volgend jaar ook nog zit!
N: En je persoonlijke wensen?
F: Het mooiste wat me kan overkomen... we zouden graag een kind willen! Nu we het er toch over hebben, nog even het slechtste: volgend jaar moet ik in dienst.
N: Beschrijf eens in vijf woorden je karakter.
F: Perseverant, blij, slordig (behalve met mijn programma's) en meer weet ik niet.
N: En hoe zit het met je favouriete films?
F: In denk Brazil en ik houd van alle Monty Python-films.
N: Wat houdt je niet van?
F: Hypocriet zijn, luie mensen, stomheid.
N: Welke interessen en hobby's heb je verder?
F: Skien, squashen, tennissen, piano spelen, science fictionverhalen en stripboeken lezen, zeilboten, vakantie, strand en zon.
N: Waar zou je het liefst heen willen gaan?
F: Noord-Amerika en Azie.
N: Wat was het beste nieuws dat je ooit te horen kreeg?
F: Dat er van STOS 35.000 exemplaren verkocht waren!!

N: En het slechtste?
F: Dat er van STOS maar 35.000 exemplaren verkocht waren!!!!

AMOS - De Amiga

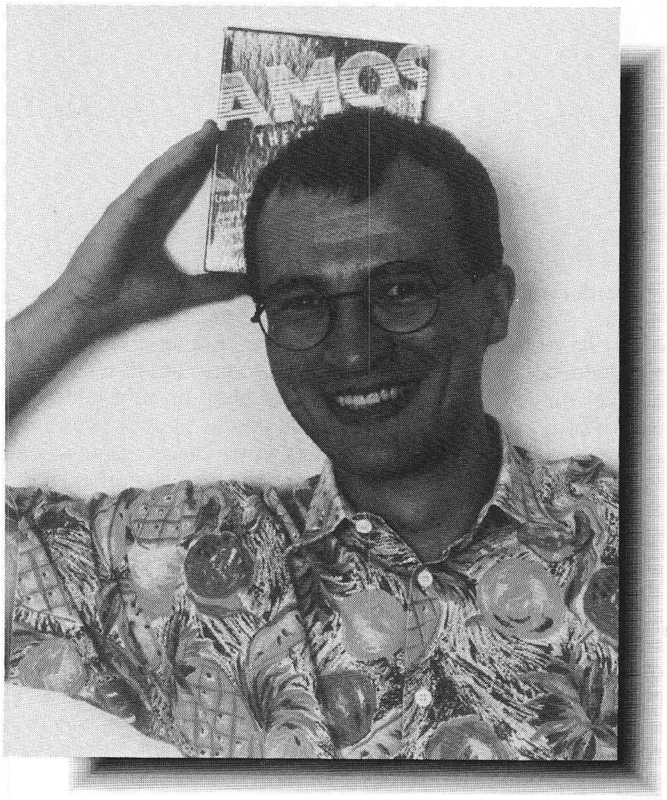
N: Wat is je favouriete programmeer-onderwerp?
F: Sprites, bobs, eigenlijk alles dat zichtbaar en hoorbaar gemaakt kan worden.
N: En daarna?
F: Onderwerpen die met sorteren te maken hebben en het maken van structuren.
N: Welke andere programmeurs bewonder je?
F: Jez San, omdat hij een groep om hem heen heeft gevormt. En anderen zoals Yves Lamoureux (van Captain Blood op de AmstradCPC)
N: Nu even over de Amiga. Wat vindt je de beste en de slechte eigenschappen van deze computer?
F: Het allerbeste zijn de chips, echt! Het slechtste is Amiga-DOS en het feit dat het C-georienteerd is. Als je machinetaal wil gebruiken, succes met de pointers!
N: Wat is je mening over het kopieren van software?
F: Het kopieren is nu een paar jaar aan de gang. Ik denk dat daardoor over vijf jaar alles op cartridges zal staan. We maken de laatste jaren mee van de micro computers! Piraten hebben de eerste wedstrijd gewonnen, maar de software-schrijvers zullen de tweede winnen.
N: Voor welke programma's was je nog meer verantwoordelijk?
F: Je kent er vast wel een paar:
Chicken Chase, Serenade, Katuvu en Ole! op de Commodore 64.Verder deed ik natuurlijk de STOS-produkties voor de Atari ST en een aantal zaken op de PC. En nu dan AMOS voor de Amiga.
N: Wat was voor AMOS je beste prestatie?
F: STOS natuurlijk. Ik werkte er een jaar aan.
N: Hoe vond je de Amiga nadat je zolang met de ST had gewerkt?
F: In het begin was ik radeloos. Ik dacht dat ik hem nooit onder de knie zou krijgen; ik haatte het apparaat gewoon.
Langzamerhand begon ik de opzet door te krijgen en ging van de chips houden, maar nog steeds haatte ik de ingebouwde software.
Uiteindelijk begon ik die ook door te krijgen, vooral door de goede documentatie.

ongeveer 50 keer diskje-verwisselen nodig om het voor elkaar te krijgen. Ik had zelfs drie dagen nodig om mijn hard-disk te installeren! Ik kan me niet voorstellen hoe de gevorderde gebruiker hiermee kan werken?!

"Ik vraag me trouwens af hoe de Amiga er in het jaar 2010 uit zal zien (omdat de Amiga er dan nog is en ST's dan niet meer bestaan)"

Op dit moment ben ik eindelijk overtuigd dat de Intuition een goed stukje vakwerk is, maar aan Amiga-DOS heb ik nog steed een hekel. Het is werkelijk zonde dat ze geen goede disk-acces in deze machine hebben gebouwd. Verder snap ik ook niet dat ze niet een paar residente programma's hebben ingebouwd. Niet veel, alleen maar DIR, COPY of iets dergelijks. Het kwam vaak voor dat ik met een diskdrive zat en geen startup-sequence op de disk had staan. Dan waren

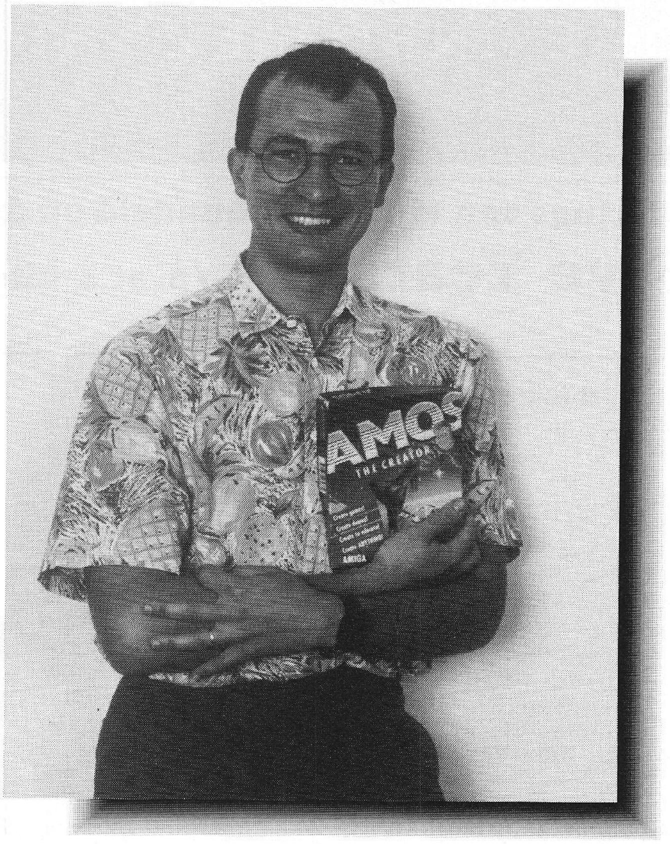
Dus mijn mening over de Amiga is een beetje in balans gekomen. Aan de ene kant prachtige hardware, maar een vreselijk slechte disk-acces en toch ook weer redelijke ingebouwde software. Verder is de Amiga gebouwd op uitbreiden. Ik vraag me trouwens af hoe de Amiga er in het jaar 2010 uit zal zien (omdat de Amiga er dan nog is en ST's dan niet meer bestaan).



AMOS - Het programma

N: Nou, wat ga je ons allemaal over AMOS vertellen?
F: Ik ga in ieder geval niet op de details in zoals in de advertenties van Mandarin.
N: Wat is er nu zo speciaal aan AMOS in vergelijking tot andere programmeertalen?
F: Het gebruik van de Intuition-interface niet. Daardoor was ik behoorlijk vrij om een geheel eigen omgeving te ontwikkelen.
AMOS is op SCREENS gebaseerd, die net niet tot de boden van het scherm rijken. Daardoor kun je zien wat er onder zit. Intuition laat dat niet toe. Ik kan het weten, wat het lukte niet om dat zote programmeren. Natuurlijk kun je per screen een andere resolutie hanteren.
N: Wat zeg je tegen mensen als ze zeggen dat je geen multi-tasking gebruikt?
F: Ik hoor veel mensen zeggen: Hij respecteert de Amiga niet, dat is toch zonde. Maar niets is minder waar. Je ziet er alleen niets van.
N: Hoe heb je het voor elkaar gekregen om zoveel functies in een programma te stoppen voor een Amiga met maar 512 KB?
F: Doordat het geen gebruik maakt van de WorkBench spaar je al 42k geheugen. En omdat alles in AMOS zelf zit, hou je nog ongeveer 240k over om mee te werken.
N: Wat kun je allemaal met AMOS programmeren?
F: Met AMOS kun je bijna alles voor elkaar krijgen. Maar het is vooral bedoeld voor het maken van spellen. Dat neemt niet weg dat je er ook databases mee kunt maken, of teksverwerkers. Maar voor deze software is AMOS waarschijnlijk te langzaam. Daarvoor kun je beter GFA of C gebruiken.
N: Welke functies zitten in AMOS die feitelijk geen enkele andere programmeertaal heeft?
F: Ik heb werkelijk al mijn stoutste dromen bij AMOS in

vervulling doen gaan, zoals een snelle editor, snelle schermen, gemakkelijke sprites, muziek, een professioneel menu-systeem, direct laden van IFF-files en meer dan een programma in het geheugen. Het is precies wat ik wilde.
N: Heb je enig idee hoe AMOS zal worden verkocht?
F: Ik weet niet hoeveel we precies van plan zijn te produceren. Laten we even van STOS uitgaan, daarvan zijn er 35.000 verkocht. De Amiga-markt is twee keer zo groot als die van de ST en AMOS is het enige programma in zijn soort. Dus je begrijpt dat ik behoorlijk optimistisch ben. En niet te vergeten alle positieve reacties die ik al had op de pre-releases... De Amiga-kring is veel vriendelijker dan die van de ST!
N: Wat doet AMOS wat geen ander kan?
F: Ik denk dat het vooral de kracht direct uit de chips haalt. Het is snel (niet zo snel als GFA, maar bijna wel), heeft meer dan 400 commando's en noem maar op.
N: Hoeveel heb je van STOS gebruikt om AMOS te maken?
F: Het probleem met mij is een beetje dat ik van te voren niet goed kan zeggen hoe een programma er uit zal gaan zien. Voordat ik met AMOS begon, kon ik alleen zeggen dat het er mooi uit zou zien, maar niet hoe. De ideeen komen pas goed als ik eenmaal ben begonnen. Maar hoe het ook zij, het programma is beter dan ik hoop had durfen dromen.
Mandarin wilde eerst een aftreksel van STOS, met de snelheid van de Amiga. Maar uiteindelijk is het een totaal ander pakket geworden met een betere programmeeromgeving en meer dan perfecte functies, dat alle voordelen van de Amiga hardware benut.
N: Hoe lang duurde de ontwikkeling van AMOS?
F: Ik had er een jaar voor nodig. Daar zaten ook de hulpprogramma's bij, zoals de Music Designer, de AMAL editor en de Sprite designer. Ik begon in april vorig jaar.
N: Was je blij met het succes van STOS?
F: Natuurlijk was ik blij met STOS. Nu begint trouwens het succes in Frankrijk door te komen. In Engeland was dat al hetgeval, maar daar merkte ik niets van. Dus voor mij was dat succes niet altijd even duidelijk merkbaar. Er was ook niemand die mij als programmeur kende. Als ik iemand vertelde dat ik programmeur was en dat ik STOS had geschreven, zeiden hij: Wat? SOTS? TSOS? Dat frustreerde me best wel.
Maar nu gaat het dus beter. Met AMOS zal het wel dezelfde kant op gaan, omdat er niet zoveel Amiga in Frankrijk zijn verkocht. Het is op zich leuk te weten dat je naam over heel de wereld wordt genoemd!



"KOOP AMOS.
Zelfs als je geen Amiga hebt"

AMOS - De toekomst

N: Welke projecten heb je verder nog in de pen?
F: Eerst militaire dienst, maar daarna wordt het weer programmeren geblazen.
N: En de plannen voor AMOS?
F: Om te beginnen de compiler en dan AMOS versie 1.1, die alle klachten van de gebruikers tot dan toe moet oplossen. En dan STOS Pro op de ST. Dat wordt dan de AMOS-versie voor de ST!
N: Als je nou niet was gaan programmeren, wat zou je dan zijn gaan doen?
F: Ik denk dat ik dan dierenarts zou zijn geworden.
N: Heb je nog iets tegen de lezers te zeggen?
F: Iedereen kan ergens de AMOS-demo disk tegenkomen; het is PD.Hoe het ook zij jullie hebben de beste machine (als ik naar m'n ST kijk wordt ik gek. Hoe kon ik stood programmeren, een heel jaar lang???)
N: Heb je verder nog iets te zeggen tegen andere programmeurs?
F: WERK! Denk niet dat je de beste bent. Alle programmeurs zijn erg trots als ze eindelijk hun programma af hebben. Was ik ook. En dan onmoette ik iemand die werkelijk echts iets moois had gemaakt. Dan voelde ik me weer heel klein.
Geloof me, je weer niet half hoe goed sommigen zijn. Ga niet naast je schoenen lopen, het kan altijd beter!
N: Wat is je boodschap tot de Amiga-wereld?
F: Ik zou tot de WERELD willen zeggen: KOOP AMOS. Zelfs als je geen Amiga hebt. Je hebt het gewoon nodig.
N: En wat tot de kopers?
F: Hartelijk dank. Koop straks de compiler!

Dit interview werd afgenomen door Martyn Brown, Newsflash UK.
Vertaling: Wilbert Scheer

CDTV

Kleintje multimedia in CD-ROM Amiga

Stop een CD-ROM, Enhanced Chip-Set en een Amiga 500 in een CD- speler kast en wat krijg je dan? Volgens Commodore CDTV, Commodore Dynamic Total Vision, het nieuwe consumenten video- en audio-gadget voor de jaren negentig. Nu staat deze CDTV-machine nog deels op de werkband en stond zij slechts onder de "glazen stolp" op CES en Comdex. Volgens Commodore zullen de eerste PAL- realeases rond Sinterklaas in de winkel staan. Wij van Commodore- Info geven alvast een vooruitblik.

Een CD-ROM drive in de Amiga stoppen is niet nieuw. Al enige tijd zijn losse CD-ROM-afspelers voor de Amiga verkrijgbaar en werd er hier en daar een Amiga 2000 of 2500 met ingebouwde CD-ROM-drive gesignaleerd. Wel nieuw is het bouwen van een multimedia Amiga in het klein rondom een gecombineerde CD-video- en CD-audiodrive. Commodore noemt dit concept Commodore Dynamic Total Vision, kortweg CDTV.

Uiterlijk

Wie de nieuwe CDTV-speler voor het eerst aan de voorzijde bekijkt raakt in vertwijfeling. Op het eerste gezicht lijkt het om een geavanceerde CD-speler te gaan of een compacte videorecorder. Zelfs de moderne goukleurige brede ronde pootjes zitten er onder en er zit een LED-venster in het frontje. Maar ...., er zit ook een 3.5 inch diskdrive in?
Aan de achterzijde lijkt het apparaat meer op een PC-systeemkast. Er zijn connectoren voor parallel Centronics, RS232 serieel, MIDI, twee audio-tulpbussen en, indien er een videokaart in de kast zit, ook een video-uitgang (composiet Pal en Y/C). Ook de bediening is voor velerlei uitleg vatbaar. Er zitten een zowel een infrarood-afstandsbediening als een muis en toetsenbord bij. Verder zitten er nog een aantal bedieningsknoppen voor de CD-ROM-drive aan de rechtervoorzijde. Dit is toch wel een erg vreemde gast in de video/audiokast.

Toch een Amiga!

Om alle twijfel weg te nemen. Het gaat wel degelijk om een Commodore Amiga. Namelijk eentje van het 500-type met een 7.16 MHz (voor PAL-applicaties op 7,09379 MHz) Motorola MC 68000, Enhanced ChipSet, 1 MB RAM, Amiga Kickstart 1.3 ROM en een video- uitbreidingsslot in. Er is een optioneel expansie-module leverbaar voor het aansluiten van een harddisk e.d.

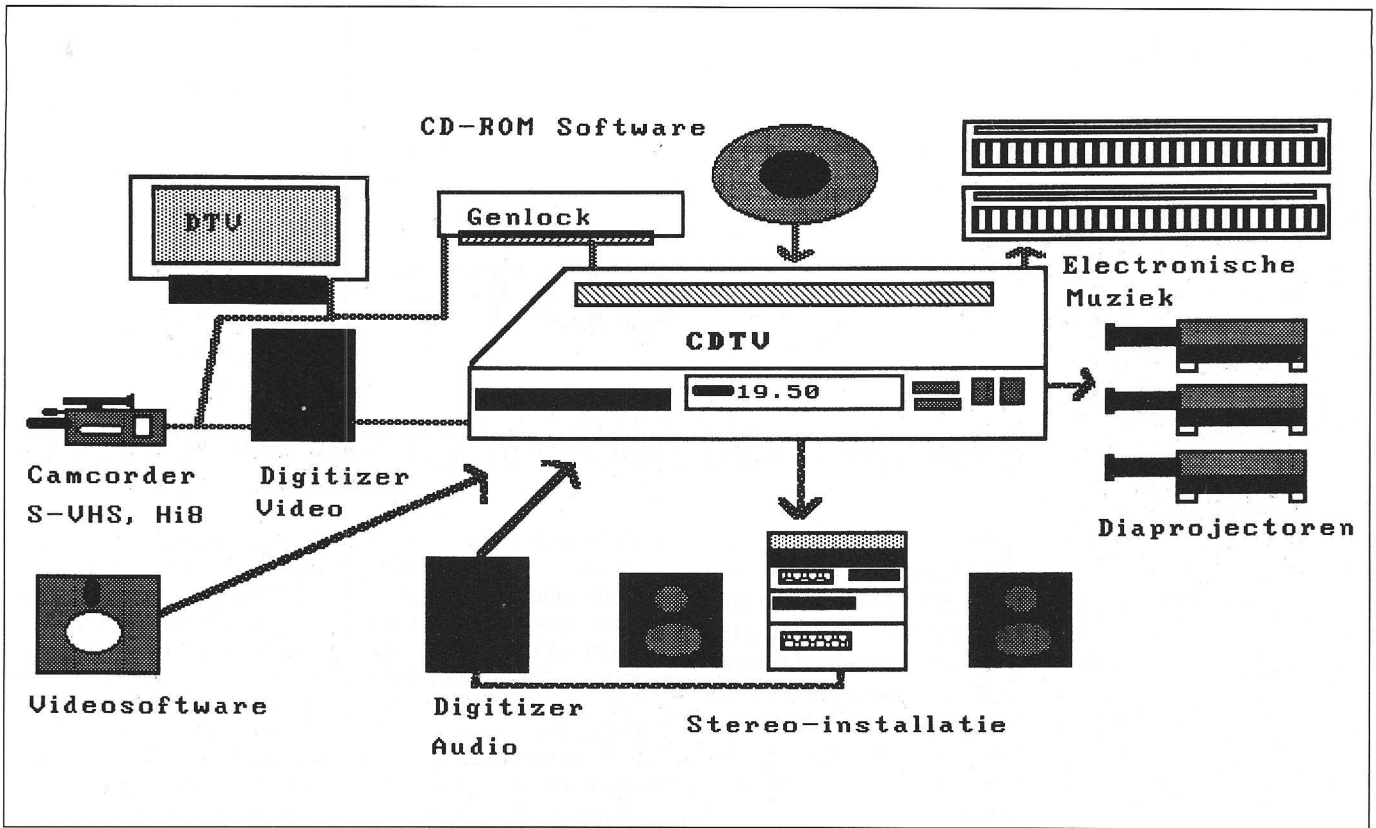
In tegenstelling tot een normaal Amiga wordt dit CDTV-model geheel Ifrarood bestuurd. Niet alleen de afstansbediening is IR. Ook de muis, het keyboard, de joysticks en tracballs zijn IR. Er is tevens een IR-tweespeler-interface. Desgewenst kunnen echter wel de conventionele muis-, joystick- en toetsenbordaansluitingen benut worden.
Het besturingssysteem Kikstart 1,3 wordt ondersteund door de ISO 9660 Fule-System-Handler en High Speed data-decompressie.

Het CD-ROM

Naast het computerhart zetelt het CD-ROM. Het gaat om een half GigaByte (550 MegaByte)-type met een Sony/Philips loopwerk korrensponderend met de CD-ROM-standaard Mode 1 en Mode 2.
De data-overdrachtssnelheid hangt af van de gebruikte modus:

- Mode 1; 153 KiloBytes per seconde
- Mode 2; 171 KB/s
- Burst-modus; 2 MegaBytes per seconde

De gemiddelde accestijd ligt tussen 0,5 en 0,8 seconde.
Deze CD-ROM-drive is zowel als CD-ROM als voor audio- en videoweergave geschikt. U kunt deze CDTV-kleine dus als encyclopedie



op schijf, geavanceerde compactdisk en videodisk- afspeler gebruiken. De toepassingen zijn legio. Van eenvoudig voorgeprogrammeerd afspelen en interactieve studiesessies (Computer Ondersteund Onderwijs, COO) tot desktopvideo en multimediapresentaties op het bedrijf of in de huiskamer. Commodore zelf spreekt zoals gezegd van Total Vision.

Toepassingen

Er is veel aandacht voor video-aansluitingen. Van huis uit is de CDTV-machine geschikt voor PAL composietvideo, Super-VHS en HI8. Optioneel is een Genlock voor dtv-toepassingen verkrijgbaar. Titel-, Paint- en animatiesoftware is reeds jaren in alle soorten en maten leverbaar. Een keurige aftiteling, tekenfilmpje of zakelijke presentatie is op deze CDTV-500 zo gepiept.
Natuurlijk is de CDTV-speler ook als gewone compactdisk bruikbaar. Via de beide tulpaansluitingen kunt u de CD via elke normale stereoversterker weergeven. Interessant is de MIDI-bus. Hiermee zijn zowel MIDI-compatibile electronische muziekinstumenten als allerlei andere randapparatuur aan te sturen. Bij die muziekinstrumenten kan de gebruiker naar hartelust componeren op de Amiga en/of met de CD-ROM AV combineren. Via de seriële en de MIDI-poort kunt u bijvoorbeeld ook één of meerdere diaprojectoren aansturen. Er zijn diverse overvloei-apparaten (dissolvers) in de handel waar het CD-geluid prima bij past U kunt de gehele diashow dan op geleide van de compactdisk programmeren.
Voor de echte multimediatoepassingen kan de gebruiker inmiddels terecht bij diverse authoring-systemen. Commodore zelf levert de auteurstaal Amiga Vision en verder is er o.a. VISA in amateur- en professionele uitvoering. Wie iets minder professioneel te werk wil gaan kan bijvoorbeeld met Deluxe Video III ook een aardige multimedia-presentatie in elkaar sleutelen.
Het "encyclopedische deel" is sterk afhankelijk van de CD-ROM- software die in de nabije toekomst leverbaar moet worden. Een CD- ROM persen en coaten kost veel geld en de vraag rijst of deze investering voor het nederlandse taalgebied er uitkomt. Algemenere CD-ROM-pakketten zoals atlassen, kunst- en historische collecties, toeristische informatie en diverse lespakketten zullen er best wel komen. Hetzelfde geldt voor met videogames en -clips beladen CD-ROMs. Een nederlandstalige encyclopedie is echter een ander kapitel. De tijd zal het leren over er ook nederlandse talige educatieve en interactieve software voor de huiskamer gaat uitkomen.
Last but not least kan de nieuwe trotse CDTV-500-bezitter dit machien natuurlijk gewoon als een Amiga 500 gebruiken. En wat daar allemaal niet mee kan...

Is CDTV en nieuw gat in de markt? Volgens Commodore gaat deze huiskamer- en kleine bedrijfsmultimediavorm het helemaal maken. Als straks de eerste kant-en-klare modellen (N.B.: De hier genoemde technische specificaties kunnen zich dus nog wijzigen!) in de winkel staan nadert het uur der waarheid. Aan het prijskaartje van minder dan *f* 2.000,- kan het in ieder geval niet liggen.

U.S.

DE TWEEDE PARALLELLE INTERFACE

Zelfbouw en achtergrondinformatie

Het maken van een extra parallelle interface is een wens van veel Amiga-bezitters. Daar willen we in dit artikel graag aan tegemoet komen. Maar voordat we daartoe overgaan eerst wat gegevens van het binnenste van de computer.

De 8520 is een randchip van het type CIA (Complex Interface Adapter, veelzijdige interface eenheid): de ontwikkelaars hebben zoveel mogelijk functies ondergebracht in een component.

De 8520

De 8520 is afgeleid van de 6526 chip die ook in de C64 is genesteld. De 8520 beschikt over 8 bit registers voor in- en uitgangen (I/O) met een handshake signaal. Beide IC's hebben twee 16 bits timers. Een seriele poort voor in of uitgang en een 24 bits teller inclusief overflow. In de Amiga zitten twee 8520's. De ene 8520 verzorgt de even adressen en de ander de oneven adressen.

De A8520 (oneven adressen) regelt het volgende:

- controle lijnen parallelle poort;
- controle lijnen seriële poort;
- een gedeelte van de diskdrive;
- Synchronisatie van de BLITTER met beeldscherm opbouw (raster).

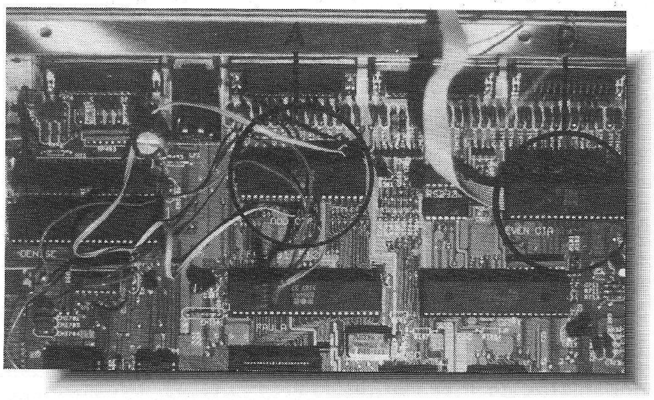
De B8520 (even adressen) regelt het volgende:

- power LED;
- een gedeelte van de diskdrive;
- de vuurknop van de joystick (poort 0 en 1);
- de data lijnen van de parallel poort;
- de controle lijnen van de parallel poort;
- keyboard (data + klok);
- klok puls (realtime klok).

De 8520 heeft de volgende mogelijkheden:

- twee vrij programmeerbare 8-bits parallelle poorten (PA en PB);
- twee 16-bits tijdtellers (timerA en timerB);
- een bidirectionele seriële poort (SP);
- een 24-bits teller, die waarschuwt bij het bereiken van een gekozen waarde (event-counter).

Al deze taken kunnen een interrupt veroorzaken. De 8520 beschikt over zestien registers. Voor de processor lijken



Registerconfiguratie van de 8520:

Register		Naam	Functie
0	$0	PRA	Gegevensregister van poort A
1	$1	PRB	Gegevensregister van poort B
2	$2	DDRA	Gegevensrichting-register van poort A
3	$3	DDRB	Gegevensregister-register van poort B
4	$4	TALO	TimerA, 8 laagste bits
5	$5	TAHI	TimerA, 8 hoogste bits
6	$6	TBLO	TimerB, 8 laagste bits
7	$7	TBHI	TimerB, 8 hoogste bits
8	$8	Event-Lo	Teller bits 0-7
9	$9	E.8-15	Teller bits 8-15
10	$A	Event-Hi	Teller bits 16-23
11	$B	---	Niet in gebruik
12	$C	SP	Gegevensregister van de seriële poort
13	$D	ICR	Interruptstuurregister
14	$E	CRA	Stuurregister A
15	$F	CRB	Stuurregister B

Parallel poort:

Register	Naam	D7	D6	D5	D4	D3	D2	D1	D0
0	PRA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
1	PRB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
2	DDRA	DPA7	DPA6	DPA5	DPA4	DPA3	DPA2	DPA1	DPA0
3	DDRB	DPB7	DPB6	DPB5	DPB4	DPB3	DPB2	DPB1	DPB0

dat normale geheugen adressen. In een systeem met een 68000 processor is alle randapparatuur namelijk 'memory mapped', dat wil zeggen de registers van de aanhangende chips doen zich voor als geheugenadressen die met de gewone instructies (bijvoorbeeld MOVE) gelezen en beschreven mogen worden.

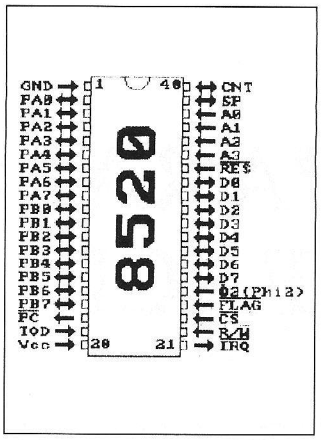
Omdat de 6526 is ontwikkeld voor processoren uit de 65xx-familie (8 bits) en de 8520 daarvan is afgeleid moet de 68000 hem aanspreken in de synchrone mode.

De E-klok van de 68000 is dus verbonden met de Phi2-ingang van de 8520. De vier adresingangen A0 tot en met A3 bepalen het actuele interne register. Van de zestien registers wordt nummer 11 ($B) niet gebruikt. De onderstaande tabel geeft een overzicht van het gebruik van de registers:

Bij de twee 8-bits parallelle poorten (PA en PB) van de 8520 horen de gegevensregisters PRA en PRB (PR, Port Registers). De chip beschikt dus over zestien poortlijnen: PA-PA7 en PB0-PB7. Elk van deze lijnen kan dienen als ingang en als uitgang (gegevensrichting). Bij de 8520 kunt u de richting van elke lijn apart definieren. Die keuze is opgeslagen is een gegevensrichting-register bij elke lijn: DDRA of DDRB (DDR, Data Direction Register). Als een bit van het gegevensrichting-register 0 is, werkt de bijbehorende lijn als ingang: lezen van de bits in het gegevensregister geeft informatie over de status van een lijn.

Analoog geeft een 1 in het gegevensregister van een lijn die lijn de status van uitgang: het nivo van de bijbehorende poortlijn komt direct overeen met de bit van het gegevensregister.

Als algemene regel geldt dat bij schrijven een waarde in het gegevenregister wordt opgeslagen; bij lezen komt



Fif 2: Pinbelegging van de 8520

de status van de poortlijnen op de databus. De bits van het gegevensrichting-register bepalen of de waarde van het gegevensregister op de poortlijnen wordt afgebeeld. Dat heeft tot gevolg dat bij lezing van een poort die als uitgang is geschakeld, de inhoud van het gegevensregister naar de poortlijnen gaat, en dat bij beschrijving van een poort die als ingang is geschakeld, de waarde naar het gegevensregister gaat. Die waarde komt pas op de poortlijnen als u omschakelt naar de status voor uitvoer.

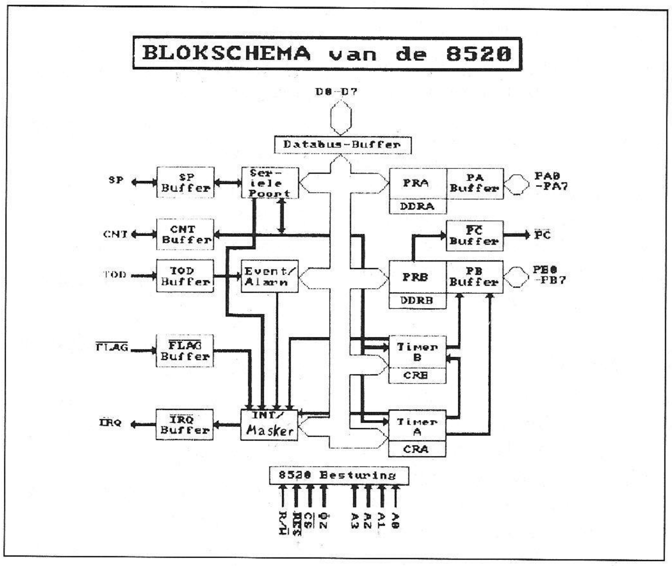
De twee handshake-lijnen (PC en FLAG) van de 8520-chips vereenvoudigen de transmissie via de parallel poort.

De PC-uitgang gaat iedere keer bij gebruik van gegevensregister B (PRB, register 1) gedurende een klokcyclus naar 0. De ingang FLAG reageert juist op negatieve flanken (dalend nivo): iedere keer dat de status van de FLAG-lijn wisselt van 1 naar 0, wordt de FLAG-bit van het interruptstuurregister gezet (ICR, register $D). Met deze twee lijnen is een eenvoudige handshake mogelijk door de FLAG- en de PC-lijnen om en om met elkaar te verbinden.

De zender hoeft zijn gegevens alleen maar in het poortregister te schrijven en op het FLAG-signaal te wachten als toestemming om het volgende byte te versturen. Omdat FLAG een interrupt mag veroorzaken, kan de zender in de wachttijd iets anders doen.

Voor de ontvanger geldt een soortgelijk verhaal.

De 8520 heeft twee 16-bits tijdtellers (timers): ze tellen vanaf een gekozen waarde terug naar nul. Ze hebben elk een groot aantal modi, te kiezen met behulp van een stuurregister (CRA en CRB). Beide timers hebben intern vier registers: twee paren (HIgh en LOw) voor twee 16-bits waarden (timerA: TALO + TAHI en PALO + PAHI). De twee paren liggen op hetzelfde adres, maar het ene kan alleen worden gelezen en het andere alleen beschreven. Bij beschrijving van een register van een timer komt de startwaarde in een latch (signaal buffer) te staan. Die waarde wordt in het telregister geschreven en



De timer (tijdteller):

Voor lezen (read)

Register	Naam	D7	D6	D5	D4	D3	D2	D1	D0
4	TALO	TAL7	TAL6	TAL5	TAL4	TAL3	TAL2	TAL1	TAL0
5	TAHI	TAH7	TAH6	TAH5	TAH4	TAH3	TAH2	TAH1	TAH0
6	TBLO	TBL7	TBL6	TBL5	TBL4	TBL3	TBL2	TBL1	TBL0
7	TBHI	TBH7	TBH6	TBH5	TBH4	TBH3	TBH2	TBH1	TBH0

Voor schrijven (write)

Register	Naam	D7	D6	D5	D4	D3	D2	D1	D0
4	PALO	PAL7	PAL6	PAL5	PAL4	PAL3	PAL2	PAL1	PAL0
5	PAHI	PAH7	PAH6	PAH5	PAH4	PAH3	PAH2	PAH1	PAH0
6	PBLO	PBL7	PBL6	PBL5	PBL4	PBL3	PBL2	PBL1	PBL0
7	PBHI	PBH7	PBH6	PBH5	PBH4	PBH3	PBH2	PBH1	PBH0

afgeteld totdat de teller op underflow (onderloop) komt; de teller staat dan op nul. De waarde in de signaalbuffer gaat automatisch opnieuw naar het telregister voor een volgend gebruik.
Lezing van een timer-register levert de actuele waarde. Het is eenvoudig in te zien dat dat alleen zin heeft als op dat moment de teller wordt gestopt. Stel dat de teller op $0100 staat. Lezing van register 5 levert het hoge byte van de actuele tellerstand: $01. Maar voordat u de kans krijgt register 4 te lezen, is er een telpuls die de tellerstand verlaagt tot $00FF. Het lage byte is nu $FF: als tellerstand wordt dus gelezen $01FF. Stilzetten van de teller is geen oplossing: tijdens de leesprocedure worden de telpulsen niet afgeteld.
De volgende procedure biedt een controlemogelijkheid: lees het hoge byte, het lage byte en nog eens het hoge byte. Als de twee waarden van het hoge byte gelijk zijn, is die waarde correct; zo niet, dan volgt een herhaling van de procedure.
In het stuurregister definieren bit 5 (timerA) en de bits 5 en 6 (timerB) welk signaal de teller verlaagt.

Bitconfiguratie van stuurregister A:

Register 14 ($E), naam: CRA

D7	D6	D5	D4	D3	D2	D1	D0
niet gebruikt	SPMODE 0=ingang 1=uitg	INMODE 0=klok 1=CNT	LOAD 1=force load strobe	RUNMODE 0=cont. 1=one shot	OUTMODE 0=puls 1=toggle	PBon 0=PB6uit 1=PB6aan	START 0=uit 1=aan

Bitconfiguratie van stuurregister B:

Register nummer 15 ($F), naam: CRB

D7	D6+D5	D4	D3	D2	D1	D0
ALARM 0=TOD 1=alarm	INMODE 00=klok 01=CNT 10=timerA 11=timerA+ CNT	LOAD 1=force load (strobe)	RUNMODE 0=cont. 1=one shot	OUTMODE 0=pulse 1=toggle	PBon 0=PB7uit 1=PB7aan	START 0=uit 1=aan

De teller (event-counter):

Register		Naam	D7	D6	D5	D4	D3	D2	D1	D0
8	$8	EventLSB	E7	E6	E5	E4	E3	E2	E1	E0
9	$9	Event8-15	E15	E14	E13	E12	E11	E10	E09	E08
10	$A	EventMSB	E23	E22	E21	E20	E19	E18	E17	E16

Bij timerA zijn maar twee bronnen mogelijk:

° 1. Het klokpuls (INMODE = 0). De telfrequentie is 716kHz, omdat de CIA's werken met de E-klok van de processor.

° 2. Het high-puls op de CNT-lijn (INMODE = 1).

Bij timerB zijn er vier modi van de ingang:

° 1. Klokcyclus (INMODE-bits = 00, binair: het eerste cijfer is bit 6 en het tweede bit 5).

° 2. VNT-puls (INMODE-bits = 01).

° 3. Onderloop van timerA (INMODE-bits = 10; de timers vormen samen een 32-bits timer).

° 4. Samenvallen van onderloop van timerA met high-nivo van de CNT-lijn (INMODE-bits = 11; biedt de mogelijkheid de lengte van een puls op de CNT-lijn te meten).

De onderloop van een teller wordt geregistreerd in het interruptstuurregister (ICR). Bit nummer 0 (TA-bit) wordt gezet bij onderloop van timerA, bit nummer 1 (TB-bit) voor timerB. Net als de andere bits van het ICR blijven deze bits gezet, totdat het register wordt gelezen. Het is ook mogelijk de onderloop van de timers uit te voeren via de parallel poort B: als in het stuurregister van een teller (CRA of CRB) de PBon-bit is gezet, verschijnt elke onderloop op de bijbehorende lijn van de poort (PB6 voor timerA en PB7 voor timerB).

Het OUTMODE-bit laat u de keus tussen twee uitvoermodi:

OUTMODE = 0 (pulse mode) Elke onderloop veroorzaakt een positieve puls op de bijbehorende lijn van de poort met de lengte van een cyclus.

OUTMODE = 1 (toggle mode) Elke onderloop veroorzaakt een wisseling van het signaal (high of low) op de lijn. Bij de start van een timer begint de uitvoer altijd met high.

Het START-bit van het ICR regelt de activiteit van een teller. De waarde 0 brengt de teller tot stilstand; de waarde 1 brengt hem op gang.
Het RUNMODE-bit kiest uit de modi 'one shot' en 'continuous'. In de eerste modus stopt de timer na elke onderloop en zet hij het START-bit terug op 0. In de continuous-mode (herhalende mode) begint de timer na de onderloop op de startwaarde.
We vermelden reeds dat een waarde niet rechtstreeks in het register van de timer

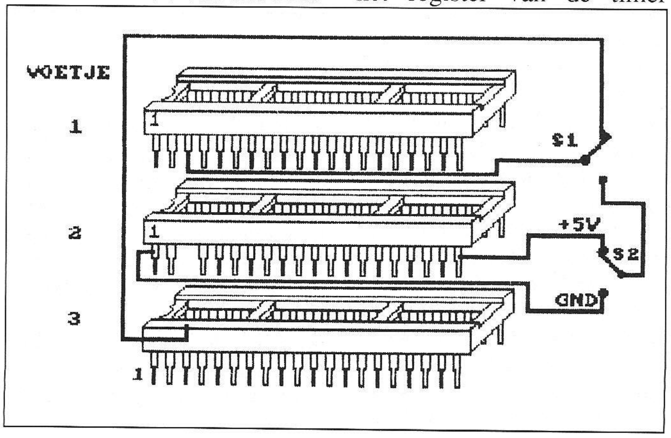


Fig. 4: De filter schakeling

wordt geschreven, maar dat dat via een signaalbuffer gebeurt. Die signaalbuffer (latch) heet ook wel prescaler (maatgever, schaalverkleiner); het aantal malen per seconde dat onderloop optreedt, is gelijk aan de telfrequentie gedeeld door de waarde in de prescaler.
De waarde in de signaalbuffer kan in drie situaties naar de teller worden verstuurd:

1. Het LOAD-bit van het ICR is gezet (force load): de waarde in de signaalbuffer gaat naar de teller, onafhankelijk van de status daarvan. Het LOAD-bit is een strobe-bit: hij wordt niet opgeslagen, maar hoeft alleen maar een handeling op gang te brengen. Voor een volgende force load moet u weer een 1 in het LOAD-bit schrijven.
2. De waarde in de signaalbuffer gaat automatisch naar de teller bij elke onderloop.
3. De waarde in de signaalbuffer gaat automatisch naar de teller na de beschrijving van het timerhigh-register, mits de teller op dat moment stilstaat (START = 0). Werkt daarom altijd met de volgorde lage byte - hoge byte.

Zoals al eerder vermeld is de 8520 een afstammeling van de 6526 uit de C64 tijd. Bij de 6526 bevond zich op deze plek een realtime clock (pen naam: Time Of Day, afgekort TOD). Commodore heeft nooit de moeite genomen om deze naam bij de 8520 te veranderen ook al heeft deze pen nu een andere functie. De huidige functie van de TOD registers is als een simpele 24 bitsteller (event counter). De 24 bits geven de event counter een bereik van 0 tot en met 16777215 ($FFFFFF). Bij iedere positieve puls op de TOD lijn (wisseling van Low naar High) gaat de teller 1 omhoog, als de event counter de stand $FFFFFF heeft bereikt springt de teller automatisch terug naar 0. U kunt de teller starten vanaf de waarde die u in de tel registers schrijft. Register 8 bevat de bits 0 tot en met 7 van de teller: ze vormen de laagste byte (LSB, Least Significant Byte). Register 9 bevat de bits 8 tot en met 15 (de middelste byte) van de teller en register 10 ($A) bevat de hoogste byte van de teller waarde (MSB, Most Significant Byte).
De teller staat tijdens het schrijven altijd stil: dat voorkomt fouten bij de overdracht van het ene naar het andere register. Pas nadat de waarde in het LSB register is geschreven, loopt de teller verder. In de meeste gevallen is het dus het beste de registers te beschrijven in de volgorde: 10 (MSB, $A), 9, 8 (LSB).
De 8520 heeft ook een 'wekker': als u van het stuurregister CRB de ALARM-bit (bit nummer 7) de waarde 1 geeft, kunt u in de registers 8 tot en met 10 een alarm waarde zetten. Zodra de waarde van de event counter gelijk is aan de alarmwaarde, wordt het ALARM-bit van het interruptstuurregister (ICR, Interrupt Control Register) gezet. In deze mode kunt u alleen schrijven in de registers. Een leespoging van de alarmwaarde mislukt: u krijgt altijd de actuele stand van de teller terug. De status van het ALARM-bit van het CRB heeft daar geen invloed op.
De seriële poort:
Register naam D7 D6 D5 D4 D3 D2 D1 D0

12 $C SDR S7 S6 S5 S4 S3 S2 S1 S0

```
(ICR, Interrupt Control Register):

Bij lezen (read): gegevensregister

Register Naam D7 D6 D5 D4   D3 D2    D1 D0
-----------------------------------------
13 $D    ICR  IR 0  0  FLAG SP ALARM TB TA

Bij schrijven (write): maskerregister
                       (sjabloonregister)

Register Naam D7   D6 D5 D4   D3 D2    D1 D0
-------------------------------------------
13 $D    ICR  S/C  x    FLAG SP ALARM TB TA
```

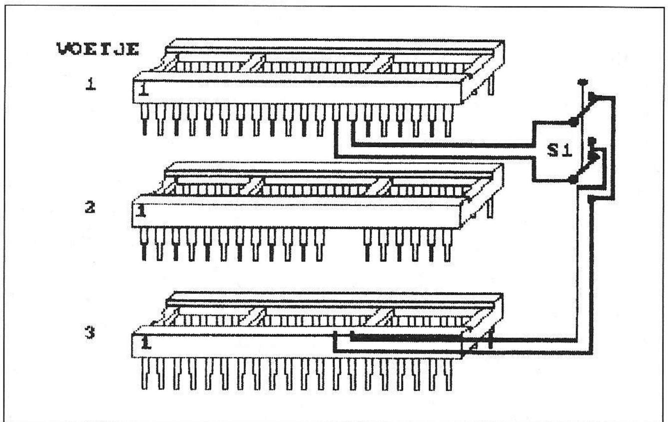


Fig. 5: De bootselector

De seriële poort bestaat hoofdzakelijk uit het seriële gegevensregister (SDR, Serial Data Register) en een 8-bits schuifregister, dat u niet zelf kunt gebruiken. Met behulp van de SPMODE-bit van het stuurregister A kiest u uit de status als ingang (SPMODE=0) en als uitgang (SPMODE=1). In de invoermode gaan seriële gegevens bij een positieve flank van de CNT-lijn naar het schuifregister. Na acht CNT-pulsen is dat vol en gaat de inhoud naar het seriële gegevensregister. Dat gaat gepaard met het zetten van het SP-bit van het interruptstuurregister (Interrupt Control Register). Als er weer CNT-pulsen komen, gaan de volgende gegevens naar het schuifregister, totdat dat weer vol is. Als u intussen het seriële gegevensregister (SDR, Serial Data Register) hebt gelezen, wordt de nieuwe waarde naar het SDR gekopieerd. De transmissie bestaat uit een voortdurend herhaling van deze cyclus.
Als SPMODE gelijk is aan 1, is de seriële poort in de uitvoermode. Het aantal keren per seconde dat timer A onderloop heeft in de continuous-mode, is maatgevend voor de baud-rate (het aantal bytes per seconde). De gegevens worden minstens in een tempo van eens per tweemaal onderloop doorgestuurd. De maximale transmissiesnelheid is een vierde van de klokfrequentie van de 8520.

Transmissie

Als de transmissie begint nadat het eerste byte is geschreven in het SDR. De CIA kopieert het databyte naar een schuifregister en de databits komen er een voor een, eens per tweemaal onderloop van timerA op de lijn SP. Het kloksignaal van timerA staat daarbij op de lijn CNT. Het nivo van die lijn wisselt bij elke onderloop van timerA, en bij elke negatieve flank (wisseling van high naar low) verschijnt het volgende bit op de lijn SP. Van een databyte wordt eerst het MSB verstuurd. Als alle acht bits zijn verstuurd, blijft het CNT hoog en de lijn SP op het nivo van de laatst uitgevoerde bit. Tevens wordt het SP-bit van het interruptstuurregister gezet; dat is het teken dat het schuifregister nieuwe gegevens mag opnemen. Als er al voor de versturing van het laatste bit een databyte in het

Adres	Naam	D7	D6	D5	D4	D3	D2	D1	D0
CIA-A: registeradressen:									
$BFE001	PRA	/FIR1	/FIR0	/RDY	/TK0	/WPRO	/CHNG	/LED	/OVL
$BFE101	PRB	Centronics parallel poort							
$BFE201	DDRA	0	0	0	0	0	0	1	1
$BFE301	DDRB	Afhankelijk van de toepassing als ingang of uitgang							
$BFE401	TALO	Het besturingssysteem gebruikt timerA voor							
$BFE501	TAHI	communicatie met het toetsenbord							
$BFE601	TBLO	Het besturingssysteem gebruikt timerB voor							
$BFE701	TBHI	diverse taken							
$BFE801	E.LSB	De event-counter van CIA-A telt de 50 Hz-impulsen							
$BFE901	E.8-15	uit het voedingsgedeelte (ticks); deze worden							
$BFEA01	E.MSB	ontleend aan de netfrequentie							
$BFEB01	SP	Opname van codes vanaf het toetsenbord							
$BFEC01	ICR	Interruptstuurregister							
$BFEE01	CRA	Stuurregister A							
$BFEF01	CRB	Stuurregister B							

gegevensregister is geschreven, gaat de transmissie zonder onderbreking verder.
Om een doorlopende transmissie te bereiken moet u ervoor zorgen dat het seriële gegevensregister op tijd wordt voorzien van nieuwe gegevens.
De SP- en de CNT-lijnen zijn uitgevoerd als open collector-uitgangen: via deze lijnen kunt u een aantal 8520's aan elkaar koppelen.

Het interruptstuurregister

Het ICR bestaat uit een gegevensregister en een maskerregister. Alle vijf interruptbronnen mogen hun bit zetten in het gegevensregister. Voor de volledigheid sommen we ze nog eens op:

1. Onderloop van timerA (TA, bit 0);
2. Onderloop timerB (TB, bit 1);
3. Gelijkheid van de waarde van de Event Counter en de alarmwaarde (ALARM, bit 2);
4. Het schuifregister van de seriële poort is vol (bij invoer) of leeg (bij uitvoer) (SP, bit 3);
5. Negatieve flank op de FLAG-ingang (FLAG, bit 4).

Als u het ICR leest, krijgt u altijd de inhoud van het gegevensregister. Dat wordt tegelijkertijd gewist: alle bits, inclusief IR, worden teruggezet! Als u verwacht de waarde in het gegevensregister nog nodig te hebben, moet u die opslaan in het RAM.
In het maskerregister kunt u alleen schrijven. De waarde bepaalt welke bits van het gegevensregister een interrupt mogen veroorzaken. Daartoe moet zo een bit de waarde 1 hebben. Zodra een bit zowel in het gegevensregister als in het maskerregister is gezet, zet de 8520 de lijn IRQ (Interrupt ReQuest) op 0: dat veroorzaakt een interrupt in de processor, omdat de lijn IRQ 0-actief is. Bovendien maakt de 8520 de IR-bit van het gegevensregister (bit 7) gelijk aan 1, zodat de interrupt ook softwarematig is terug te vinden. Pas als het ICR is gelezen en dus het gegevensregister is gewist, gaat de lijn IRQ terug naar 1.
De beschrijving van het maskerregister verloopt anders dan bij een gewone geheugenplaats: die is alleen mogelijk als tegelijkertijd het bedoelde bit en het S/C bit (Set/Clear, bit 7) worden gezet. De overige bits blijven dan onveranderd. Als u het S/C bit definieert dus of de die u in uw masker opgeeft, worden gezet (S/C = 1) of gewist (S/C = 0). De niet gezette bits in het masker hebben geen invloed op de bits in het maskerregister.
Veronderstel dat de inhoud van het maskerregister nu (binair) 00000011 is (beide timer-interrupts toegestaan). Stel dat u alleen een interrupt via de FLAG-lijn wilt toestaan. Voor deze verandering moet u 10010000 in het maskerregister schrijven: S/C = 1, zodat het gezette bit (FLAG, bit 5) het overeenkomstige bit van het maskerregister zet. Als resultaat van de bewerking heeft het maskerregister de inhoud 00010011.
Om vervolgens de timer-interrupt onmogelijk te maken gebruikt u de waarde 00000011: het S/C bit is 0, zodat de gezette bits worden gewist (nummer 1 en 0, TB en TA). Het resultaat in het maskerregister is dus 00010000. Van nu af kan alleen FLAG nog een interrupt veroorzaken.

De plaats van de 8520's

Het basisadres van de eerste CIA (8520A) is $BFE001. Zijn registers liggen niet op opeenvolgende posities, maar met tussenruimten van 256 bytes: alle adressen van de 8520A zijn oneven, omdat hij met de acht laagste lijnen van de adresbus is verbonden (D0-D7).

De tweede CIA (CIA-B) is bereikbaar via adres $BFD000. Zijn registers liggen op de even adressen, omdat de databus van CIA-B is verbonden met de bovenste helft van de databus van de processor.

Adres	Naam	D7	D6	D5	D4	D3	D2	D1	D0
CIA-B: registeradressen									
$BFD000	PRA	/DTR	/RTS	/CD	/CTS	/DSR	SEL	POUT	BUSY
$BFD100	PRB	/MTR	/SEL3	/SEL2	/SEL1	/SEL0	/SIDE	DIR	/STEP
$BFD200	DDRA	1	1	0	0	0	0	0	0
$BFD300	DDRB	1	1	1	1	1	1	1	1
$BFD400	TALO	TimerA is alleen in gebruik bij seriële							
$BFD500	TAHI	transmissie van gegevens, verder heeft hij geen functie							
$BFD600	TBLO	TimerB synchroniseert de BLITTER							
$BFD700	TBHI	met het scherm, verder heeft hij geen functie							
$BFD800	E.LSB	De event-counter van CIA-B telt de impulsen							
$BFD900	E.8-15	van de horizontale synchronisatie; in het							
$BFDA00	E.MSB	PAL-formaat zijn dat er 15625 per seconde							
$BFDB00	SP	Niet in gebruik							
$BFDC00	ICR	Interruptstuurregister							
$BFDE00	CRA	Stuurregister A							
$BFDF00	CRB	Stuurregister B							

A8520

pen	functie	verklaring:
PA0:	BUSY	parallel poort uitgang met SP verbonden
PA1:	POUT	parallel poort ingang met CNT verbonden
PA2:	SEL	parallel poort ingang
PA3:	-DSR	seriële poort ingang via buffer MC1489A
PA4:	-CTS	seriële poort ingang via buffer MC1489A
PA5:	-CD	seriële poort ingang via buffer MC1489A
PA6:	-RTS	seriële poort uitgang via buffer MC1488
PA7:	-DTR	seriële poort uitgang via buffer MC1488
PB0:	-STEP	drive signaal step (track opschuiven)
PB1:	DIR	drive signaal richting
PB2:	-SIDE	drive signaal side-select (boven of onderkant)
PB3:	-SEL0	drive signaal drive 0
PB4:	-SEL1	drive signaal drive 1
PB5:	-SEL2	drive signaal drive 2
PB6:	-SEL3	drive signaal drive 3
PB7:	-MTR	drive signaal motor aan
SP:	BUSY	parallel poort uitgang met PA0 verbonden
CNT:	POUT	parallel poort uitgang met PA1 verbonden
-PC:	niet in gebruik	
-FLAG:	-INDEX	drive signaal disk index
1*TOD:	-BHS	Time Of Day
TIMERA:	niet in gebruik	
TIMERB:		wordt gebruikt voor synchronisatie van de blitter met het beeldscherm

B8520

pen	functie	verklaring
PA0:	OVL	Memory Overlay
PA1:	-LED	Power-LED kontrole
PA2:	-CHNG	drive signaal Diskchange
PA3:	-WPRO	drive signaal Disk write protect
PA4:	-TK0	drive signaal Track 0
PA5:	-RDY	drive signaal Disk-Ready
PA6:	-JOY0	Joystick poort 0 pen 6 (vuurknop)
PA7:	-JOY1	Joystick poort 1 pen 6 (vuurknop)
PB0-PB7:	-P0--P7	parallel poort data lijnen
SP:	KDAT	Keyboard data
CNT:	KCL	Keyboard clock
-PC:	-DRDY	parallel poort ingang strobe
-FLAG:	-ACK	parallel poort uitgang Acknowledge (bevestiging)
1*TOD:	TICK	Time Of Day
TIMERA:		door het toetsenbord in gebruik
TIMERB:		virtuele Timer-Device voor Multitasking en Interrupts

Commodore geeft de adressen $BFD000 en $BFE001 op als basisadressen van CIA-B respectievelijk CIA-A. Bij een nauwkeurige beschouwing van het schema ontdekken we dat de twee CIA's mogen worden geadresseerd in het gehele gebied van A0xxxx tot BFxxxx. De adreslijnen A12 en A13 activeren de CIA's. Mits het adres ligt tussen A0xxxx en BFxxxx, activeert A12 = 0 de CIA-A en geldt A13 = 0 voor CIA-B. Omdat de databus van CIA-A met de lijnen D0-D7 (oneven adressen) van de databus van de processor is verbonden en CIA-B met D8-D15 (even adressen), kunt u ze, met een woord, tegelijk aanspreken (mits A12 en A13 beide 0 zijn).

De opdracht 'MOVE.W $BF0000, D0' laadt de PA-registers van de twee CIA's in D0: de acht laagste bits van D0 bevatten dan de inhoud van het PA-register van CIA-A en bit 9-15 die van het PA-register van CIA-B.

Uit het voorgaande volgt een schema voor de adressering van een CIA. Kies CIA-A met behulp van (binair):

```
101x xxxx xxx0 rrrr
    xxxx xxx1
```

CIA-B met:

```
101x xxxx xx0x rrrr
    xxxx xxx0
```

De vier bits 'rrrr' definieren het gewenste register.

De pinbelegging van de 8520 is zichtbaar in **figuur 2**. Hier onder volgt een overzicht van wat de verschillende pennen van de 8520 doen. Dit overzicht is verschillend voor de A en de B versie van de 8520. De '-' voor een afkorting betekend dat de desbetreffende data lijn laag aktief is.

1*TOD: Deze afkorting staat voor Time Of Day. Deze term stamt nog af van de voorganger van de 8520, namelijk de 6526. Commodore heeft nooit de moeite genomen om dit te veranderen. Voor de huidige betekenis moet men hierboven in de tekst kijken

De parallel poort

De parallel poort is de poort die meestal wordt gebruikt voor de printer. Een printer wordt ook wel aangestuurd via de serielepoort maar deze is lang niet zo snel. Verder wordt de parallelpoort van de Amiga onder andere gebruikt voor de volgende toepassingen:

- ° printer
- ° sound sampler
- ° video digitiser
- ° I/O kaarten
- ° scanner
- ° joystick (voor Gauntlet e.d.)
- ° dongles (sleutels voor beveiligde software)
- ° fax

Besturing

De parallel poort wordt bestuurt door de CIA 8520 chips. Deze chips hebben al een eigen reputatie opgebouwd wat betreft het stuk gaan. Indien uw Amiga het niet meer doet (of een van de connectors werkt niet meer) is er een kans van 90% dat een van de 8520 chips kapot is.

Om te controleren of er een 8520 kapot is raden we aan om bij een vreemde (de buurman?) Amiga de 8520's uit hun voetjes te wippen en deze in uw eigen favoriete Amiga te prikken. Denk er

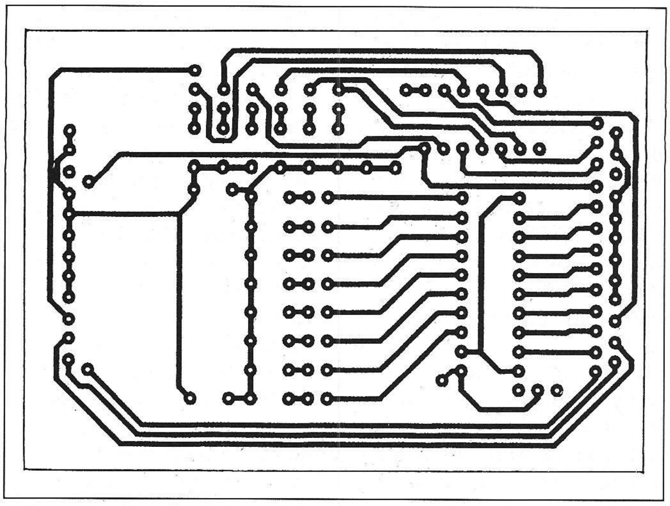


Fig. 6: Onderzijde print in spiegelbeeld

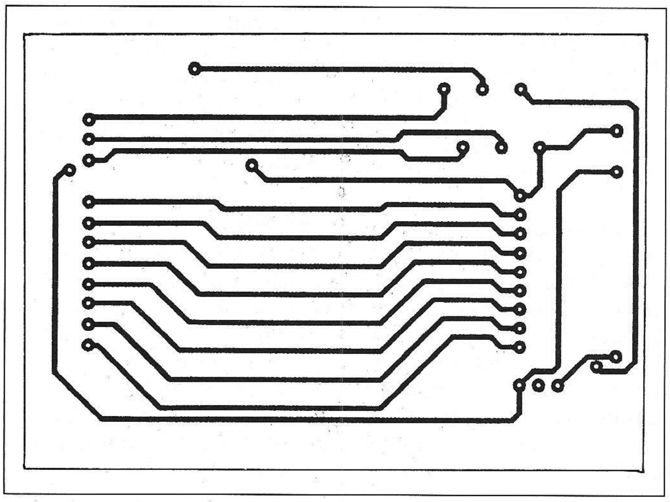


Fig. 7: Bovenzijde print

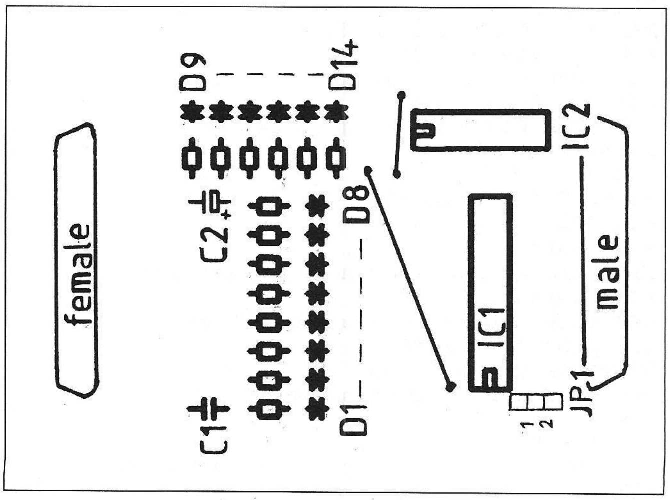


Fig. 8: Componenten opstelling

Benodigde componenten:

```
* IC1        SN 74HCT540
* IC2        SN 74HCT04
* D1-D8      LED 3mm kleur rood
* D9         LED 3mm kleur geel
* D10-D14    LED 3mm kleur groen
* R1-R14     330 ohm
* C1         100 nF
* C2         100 mF/25V staand model
* JP1        3 polige pennen strip en een codeerbrug
*            SUB D connector haaks print montage female
*            SUB D connector haaks print montage male
*            geïsoleerd draad voor de beide
                  draadbruggen
```

bij het uithalen van de kostbare chips (Fl 55,)) erom dat de pootjes niet ombuigen. Dit kan men voorkomen door het IC er met een klein schroevedraaiertje voorzichtig uit te wippen (gelijkmatig aan beide kanten).
Door nu een beetje te wisselen met de kapotte en de goede 8520's heeft uw een goed werkende Amiga en de ander een kapotte.

Enkele voorbeelden

Om nu het gebruik van de 8520's te demonstreren, hebben we twee kleine bouwprojecten en een grote voor u bedacht.
Het zal menigeen duidelijk zijn dat het bij de huidige Amiga's mogelijk is het geluids filter te schakelen door de POWER-LED aan of uit te schakelen. We konden dit al softwarematig doormiddel van:

BASIC:

```
POKE 12574721, 2
   ;LED/filter aan
POKE 12574721, 253
   ;LED/filter uit
```

Assembler:

```
MOVE.B #2, $BFE001
   ;LED/filter aan
MOVE.B #253,
   $BFE001
   ;LED/filter uit
```

In **figuur 4** kunt u het eenvoudige schema zien. Het schema is door iedereen te bouwen, het is zelfs Hans gelukt. Het principe van de LED switch is: we onderbreken gewoon het pootje van de 8520 waaraan het LEDje/filter is aangesloten en plaatsen hier een schakelaar tussen. In de ene stand is het pootje gewoon doorverbonden en functioneert het geheel normaal, en in de andere stand kunnen we met de tweede schakelaar kiezen in welke stand het LEDje/filter moet staan. Het bewuste pootje van de 8520 is pootje 3 van de 8520-A. We maken gebruik van drie extra voetjes waarvan we pootje drie van het bovenste voetje naar buiten toe buigen. Verder wordt het derde pootje van het tweede voetje geheel verwijderd. Nu kan men de bedrading eraan solderen en de schakelaars. Ten slotte drukken we de drie voetjes in elkaar en drukken we het gevaarte in de Amiga (met er bovenop de 8520).

Het tweede simpele voorbeeld is de altijd zeer bruikbare bootselector. In de winkel hebben ze het lef om hiervoor Fl30,- te vragen, maar indien u 20 minuten de tijd neemt, kan u de bootselector bouwen voor Fl9,95. De bootselector is een schakelaar die het mogelijk maakt om doormiddel van een schakelaar te kiezen vanaf welke drive moet worden geboot (df0: of df1:). In **figuur 5** kunt u het volledige schema zien. Net als bij de LED-switch, moet men de 8520 uit zijn voetje halen om er drie voetjes onder te zetten (alleen hebben we hier te maken met de 8520 die het dichst bij de disk-drive zit (A500).
Bij dit ontwerp vouwen we

de pootjes 13 en 14 om van het IC voetje 1. Verder verwijderen we de pennetjes 13 en 14 van het tweede voetje. Nu kunnen we het geheel in elkaar solderen, let wel: de schakelaar moet dubbel polig-om zijn. Verder monteren we het bouwsel in de Amiga op dezelfde manier als bij de LED/filter switch, maar dan onder de 8520-B.

Als laatste volgt er dan nu nog een parallel monitor. Met behulp van een parallel monitor kan de gebruiker zien hoe de lijnen van de parallel poort staan (hoog of laag). Tevens is het mogelijk om met de parallel monitor te kijken of het gedeelte van de 8520 dat de parallel poort regelt kapot is.
Een van de schrijvers van dit artikel vond op deze manier waarom zijn Amiga zo moeilijk deed met het printen van een stukje tekst, het bleek dat er een 8520 kapot was.

Hoe werkt de parallel-poort monitor

De werking van dit ontwerp is vrij eenvoudig. De beide IC's werken als buffer om de LED's aan te sturen, de 8520's in de Amiga gaan beslist stuk als ze een LED rechtstreeks zouden moeten aansturen.
De print is zo ontworpen dat elke LED gaat branden als de bijbehorende data lijn actief wordt. Dit houdt in dat de LED's 9, 10 en 11 gaan branden als de spanning op de bijbehorende aansluiting laag (0V) is. De overige LED's gaan branden als de spanning hoog (+5V) is.

Welke LED geeft nu wat aan?

```
D1     datalijn 0
D2     datalijn 1
D3     datalijn 2
D4     datalijn 3
D5     datalijn 4
D6     datalijn 5
D7     datalijn 6
D8     datalijn 7
D9     -RESET
D10    -STROBE
D11    -ACK
D12    SEL
D13    POUT
D14    BUSY
```

Bouw

Voor het bouwontwerp is een dubbelzijdig geetste printplaat nodig (zie **afbeelding 6 en 7**). Deze afbeeldingen zijn in schaal 1 op 1 afgedrukt en kunt u direct gebruiken. Denk er wel om dat de onderzijde (**afbeelding 6**) in spiegelbeeld is afgedrukt en dat de bovenzijde (**afbeelding 7**) niet in spiegelbeeld is afgedrukt. Als u de printplaat heeft kunt u beginnen met de opbouw. Hierbij kan men het beste beginnen met de beide draad bruggen en de doorverbinding van de bovenzijde en de onderzijde van de print naast D9. Dan volgen de onderdelen die aan beide zijde van de print gesoldeerd moeten worden (de beide IC's, de weerstand naast D11, de beide connectors en de jumper). Als u voor de IC's voetjes wilt gebruiken kunt u alleen IC-voetjes gebruiken met professioneel gedraaide buscontacten. Voor de beide connectors heeft u SUB D connectors nodig met haakse aansluiting voor print montage. Let er wel op dat deze connectors ook van bovenaf gesoldeerd moeten kunnen worden.
Hierna kunt u de overige weerstanden staand in de print solderen. Deze zijn in **afbeelding 8** niet genummerd omdat alle weerstanden dezelfde waarden hebben en u zich dus niet kunt vergissen. Denk er bij de LED's om dat u ze goed om in de print steekt, de LED's 9, 10 en 11 zitten andersom als de LED's 12, 13 en 14! Als laatste kunt u de beide condensators monteren. Zet C2 wel goed om in de print!
Om de print voor alle Amiga's geschikt te maken is de jumper(JP1) toegevoegd. In stand 1 (zie **afbeelding 8**) werkt de print alleen op een Amiga 1000. Tevens dient voor de Amiga 1000 bij beide SUB-D connectors pen 25 recht gebogen te worden. Vervolgens moet men deze pennen met een stukje draad aan elkaar verbinden. Het LEDje -RESET zal dan altijd blijven branden. In stand 2 werkt de print met de overige Amiga's (A500, A2000, A2500, A3000). U kunt natuurlijk ook een stukje draad gebruiken als u de print alleen op uw eigen type Amiga gebruikt.
Voordat u de print aan uw Amiga aansluit moet u de print goed nakijken op eventuele fouten. Sluit de print alleen aan als de Amiga en de printer beide uit zijn.

Software

Het programma is dit keer in BASIC geschreven, zodat iedereen dit kan intikken en gebruiken. Als u het programma heeft ingetikt en naar disk heeft weggeschreven kunt u het programma starten. Op het scherm verschijnen dan 14 vierkantjes in dezelfde opstelling als de LED's op de print. Tevens verschijnen er dan nog 5 rechthoeken met de teksten: NEXT, LAST, PLAY, STOP en 0. Als u nu een van de 14 vierkantjes aanklikt veranderd deze van kleur en licht het bijbehorende LEDje op de print ook op (alleen als de print is aangesloten natuurlijk!), behalve de bovenste 3 vierkantjes, deze kunt u wel aanklikken maar de bijbehorende LED zal niet op lichten.
Als u nu een aantal van de 11 vierkantjes die wel van kleur kunnen veranderen in een bepaald patroon heeft aangeklikt kunt u NEXT of LAST aanklikken. U zult zien dat vierkantjes weer allemaal dezelfde kleur hebben en dat de 0 een 1 dan wel een 15 is geworden. U kunt nu weer een nieuw patroon maken. Als u alle 16 patronen heeft kunt u PLAY aanklikken en u zult zien dat de patronen die u heeft ingevoerd een voor een op het scherm en op de print zichtbaar worden. Door nu op de linker-muis toets te drukken stopt het weergeven van de afzonderlijke patronen en springt de teller terug op nul.

Tot slot

Dit artikel is geschreven/gemaakt door D.J. Brandt en H. van der Pol. Alle bouwprojecten zijn meerdere keren gebouwd en getest. De schrijvers en Commodore Info kunnen niet aansprakelijk worden gesteld voor eventuele fouten.

Hieronder volgt het programma in Amiga-Basic:

```
REM (c) D.J. Brandt
REM voor Commodore Info
POKE 12575489&,255
REM DDRB CIA-A op uitgang zetten
POKE 12571136&,199
REM DDRA CIA-B op uitgang zetten
adress1 = 12574977&
REM PRB CIA-A
adress2 = 12570624&
REM PRA CIA-B
POKE adress1,0
REM alle uitgangen van PRB uitzetten
POKE adress2,0
REM alle uitgangen van PRA uitzetten
DIM prb(16),pra(16)
ON MOUSE GOSUB sstop
REM wordt pas actief na MOUSE ON
CLS
FOR i = 0 TO 7
   LINE (118+i*40,130)-(138+i*40,140),1,bf
NEXT i
REM tekent horizontaal 8 vierkantjes
FOR i = 0 TO 5
   LINE (440,14+i*24)-(460,24+i*24),1,bf
```

```
NEXT i
REM tekent vertikaal 6 vierkantjes
LINE (168,26)-(208,44),1,b
LINE (148,60)-(168,44),1
LINE (228,60)-(208,44),1
teller = 0
LOCATE 5,23 : PRINT teller
LOCATE 19,16 : PRINT "D0    D1    D2    D3
    D4    D5    D6    D7"
LOCATE 3,60  : PRINT "RESET"
LOCATE 6,60  : PRINT "STROBE"
LOCATE 9,60  : PRINT "ACK"
LOCATE 12,60 : PRINT "SEL"
LOCATE 15,60 : PRINT "POUT"
LOCATE 18,60 : PRINT "BUSY"
FOR i=0 TO 3
   LINE (118+i*80,60)-(178+i*80,74),1,b
NEXT i
REM tekent horizontaal 4 rechthoeken
LOCATE 9,17 : PRINT "NEXT"
LOCATE 9,27 : PRINT "LAST"
LOCATE 9,37 : PRINT "PLAY"
LOCATE 9,47 : PRINT "STOP"
LINE (358,60)-(418,74),3,b
mousecheck:
   IF MOUSE(0)<1 GOTO mousecheck:
   CALL muis(x,y)
   REM roept de subroutine muis aan
   REM in de variabele x staat de x-positie
    van de muispointer
   REM in de variabele y staat de y-positie
    van de muispointer
   IF x>118 AND x<138 AND y>130 AND y<140
    THEN GOSUB d0
   IF x>158 AND x<178 AND y>130 AND y<140
    THEN GOSUB d1
   IF x>198 AND x<218 AND y>130 AND y<140
    THEN GOSUB d2
   IF x>238 AND x<258 AND y>130 AND y<140
    THEN GOSUB d3
   IF x>278 AND x<298 AND y>130 AND y<140
    THEN GOSUB d4
   IF x>318 AND x<338 AND y>130 AND y<140
    THEN GOSUB d5
   IF x>358 AND x<378 AND y>130 AND y<140
    THEN GOSUB d6
   IF x>398 AND x<418 AND y>130 AND y<140
    THEN GOSUB d7
   IF x>440 AND x<460 AND y>110 AND y<120
    THEN GOSUB pout
   IF x>440 AND x<460 AND y>134 AND y<144
    THEN GOSUB busy
   IF x>440 AND x<460 AND y>86 AND y<96
    THEN GOSUB sel
   IF x>118 AND x<178 AND y>60 AND y<74
    THEN GOSUB nnext
   IF x>198 AND x<258 AND y>60 AND y<74
    THEN GOSUB last
   IF x>278 AND x<338 AND y>60 AND y<74
    GOTO play
   LOCATE 1,1
   PRINT "                                    "
   IF x>440 AND x<460 AND y>14 AND y<24
    THEN GOSUB text
   IF x>440 AND x<460 AND y>38 AND y<48
    THEN GOSUB text
   IF x>440 AND x<460 AND y>62 AND y<72
    THEN GOSUB text
   POKE adress1,prb(teller)
   REM schrijft de waarde prb(teller) naar
    PRB
   POKE adress2,pra(teller)
   REM schrijft de waarde pra(teller) naar
    PRA
GOTO mousecheck
d0:
   IF POINT (120,135)=1 THEN
      REM afhankelijk van de kleur van het
      REM vierkantje wordt de waarde
      REM prb(teller) verhoogd danwel ver-
      REM laagd.
      a = 2
      prb(teller) = prb(teller) + 1
   ELSE
      a = 1
      prb(teller) = prb(teller) - 1
   END IF
   LINE (118,130)-(138,140),a,bf
   REM het vierkantje krijgt hier een nieuw
    kleurtje
RETURN
d1:
   IF POINT (160,135)=1 THEN
      a = 2
      prb(teller) = prb(teller) + 2
   ELSE
      a = 1
      prb(teller) = prb(teller) - 2
   END IF
   LINE (158,130)-(178,140),a,bf
RETURN
d2:
   IF POINT (200,135)=1 THEN
      a = 2
      prb(teller) = prb(teller) + 4
   ELSE
      a = 1
      prb(teller) = prb(teller) - 4
   END IF
   LINE (198,130)-(218,140),a,bf
RETURN
d3:
   IF POINT (240,135)=1 THEN
      a = 2
      prb(teller) = prb(teller) + 8
   ELSE
      a = 1
      prb(teller) = prb(teller) - 8
   END IF
   LINE (238,130)-(258,140),a,bf
RETURN
d4:
   IF POINT (280,135)=1 THEN
      a = 2
      prb(teller) = prb(teller) + 16
   ELSE
      a = 1
      prb(teller) = prb(teller) - 16
   END IF
   LINE (278,130)-(298,140),a,bf
RETURN
d5:
   IF POINT (320,135)=1 THEN
      a = 2
      prb(teller) = prb(teller) + 32
   ELSE
      a = 1
      prb(teller) = prb(teller) - 32
   END IF
   LINE (318,130)-(338,140),a,bf
RETURN
d6:
```

```
   IF POINT (360,135)=1 THEN
      a = 2
      prb(teller) = prb(teller) + 64
   ELSE
      a = 1
      prb(teller) = prb(teller) - 64
   END IF
   LINE (358,130)-(378,140),a,bf
RETURN
d7:
   IF POINT (400,135)=1 THEN
      a = 2
      prb(teller) = prb(teller) + 128
   ELSE
      a = 1
      prb(teller) = prb(teller) - 128
   END IF
   LINE (398,130)-(418,140),a,bf
RETURN
sel:
   IF POINT (445,90)=1 THEN
      a = 2
      pra(teller) = pra(teller) + 4
   ELSE
      a = 1
      pra(teller) = pra(teller) - 4
   END IF
   LINE (440,86)-(460,96),a,bf
RETURN
pout:
   IF POINT (445,115)=1 THEN
      a = 2
      pra(teller) = pra(teller) + 2
   ELSE
      a = 1
      pra(teller) = pra(teller) - 2
   END IF
   LINE (440,110)-(460,120),a,bf
RETURN
busy:
   IF POINT (445,140)=1 THEN
      a = 2
      pra(teller) = pra(teller) + 1
   ELSE
      a = 1
      pra(teller) = pra(teller) - 1
   END IF
   LINE (440,134)-(460,144),a,bf
RETURN
nnext:
   REM hier wordt de waarde teller met 1
    verhoogd,
   REM als de waarde teller hoger is als 15
    dan
   REM wordt teller teruggezet op 0
   teller = teller + 1
   IF teller > 15 THEN teller = 0
   LOCATE 5,23 : PRINT teller
   GOSUB schermaanpassen
   REM subroutine aanroepen om het
   REM scherm aan te passen
RETURN
last:
   REM hier wordt de waarde teller met 1
    verlaagd,
   REM als de waarde teller lager is als 0
    dan
   REM wordt teller op 15 gezet
   teller = teller - 1
   IF teller < 0 THEN teller = 15
   LOCATE 5,23 : PRINT teller
   GOSUB schermaanpassen
   REM subroutine aanroepen om het
   REM scherm aan te passen
RETURN
play:
   LOCATE 1,1
   PRINT "    Linker muis-toets om te
    stoppen                          "
   LINE (358,60)-(418,74),1,b
   LINE (278,60)-(338,74),3,b
play1:
   MOUSE ON
   REM ON MOUSE actief maken
   FOR teller = 0 TO 15
      LOCATE 5,23 : PRINT teller
      POKE adress1,prb(teller)
      POKE adress2,pra(teller)
      GOSUB schermaanpassen
   NEXT
GOTO play1
sstop:
   MOUSE OFF
   REM ON MOUSE inactief maken
   LINE (278,60)-(338,74),1,b
   LINE (358,60)-(418,74),3,b
   teller = 0
   LOCATE 5,23 : PRINT teller
   GOSUB schermaanpassen
   test = MOUSE(0)
GOTO mousecheck
schermaanpassen:
   buffer = prb(teller)
   REM de waarde prb(teller) opslaan in
    buffer.
   FOR i = 7 TO 0 STEP -1
      REM hier wordt uitgezocht
      REM welk vierkantje welke
      REM kleur moet krijgen.
      IF prb(teller) - 2^i => 0 THEN
         prb(teller) = prb(teller) - 2^i
    LINE (118+i*40,130)-(138+i*40,140),2,bf
      ELSE
         LINE
    (118+i*40,130)-(138+i*40,140),1,bf
      END IF
   NEXT
   prb(teller) = buffer
   buffer = pra(teller)
   FOR i = 5 TO 3 STEP -1
      IF pra(teller) - 2^(i-3) => 0 THEN
         pra(teller) = pra(teller) - 2^(i-3)
         LINE
    (440,14+(8-i)*24)-(460,24+(8-i)*24),2,bf
      ELSE
         LINE
    (440,14+(8-i)*24)-(460,24+(8-i)*24),1,bf
      END IF
   NEXT
   pra(teller) = buffer
RETURN
text:
   LOCATE 1,5
   PRINT "Deze led is niet software matig
    aan te sturen!"
RETURN
SUB muis(x,y) STATIC        :
   REM subroutine die aan x en y de actuele
   REM muispointer coordinaten toekent
   x = MOUSE(1)
   y = MOUSE(2)
END SUB
```

NIEUWS EN FEITEN

Power PC Board

Een aantal nummers terug namen we in Commodore Info het bericht op van de firma Rossmuller uit Duitsland die een eigen ontwikkeling zou hebben van het Power PC Board, die in Nederland door Flevosoft zou worden geleverd.
Het is gebleken dat het PC Board van Rossmuller hetzelfde is als dat van KCS uit Nederland om de eenvoudige reden dat Rossmuller ze gewoon bij KCS in Nederland heeft gekocht. Voor alle duidelijkheid: er is op dit moment maar een PC Board voor de 500 in de wereld en die is van KCS!
Overigens nemen we het volgende nummer een test op van dit board.

AMOS

Er is een nieuwe programmeertaal voor de Amiga. Dit pakket wordt in Nederland niet door Homesoft geimporteerd, maar door Amicomp.
Desgevraagt deelde Mandarin, Engeland bij monde van Jocelyn Cocker ons mee dat Homesoft niet accuraat zou hebben gereageerd op verzoeken van Mandarin het pakket zo snel mogelijk in Nederland te verspreiden. Als goed alternatief werd daarom voor Amicomp gekozen. Alle overige Mandarin software wordt normaal onder de Homesoft-paraplu gedistribueert. In de volgende nummer brengen we een test van dit programma. In dit nummer alvast een interview met de maker.
Inl: Amicomp, 05220-54043

A-MAX II

Er is een upgrade verschenen van A-Max, de Macintosh-emulator voor de Amiga. A-Max II kan de meeste harddisk controllers nu aan en ondersteunt nu ook andere SCSI-aparatuur, zoals de LaserWriter IISC en scanners. Verder werken processorversnellerkaarten nu ook met A-Max zodat de boel tot vijf keer sneller draait. Geregistreerde gebruikers krijgen binnenkort bericht over de updates.
ReadySoft werkt daarnaast ook aan A-Max II Plus, dat een Amiga 2000-kaart is, die compatible moet worden met AppleTalk en MIDI. Als bijzonderheid noemt ReadySoft dat Mac-schijfjes op een normale Amiga-drive kunnen worden gelezen. Naar alle waarschijnlijkheid is deze kaart eind oktober leverbaar (in Amerika).



AT-emulator voor 500

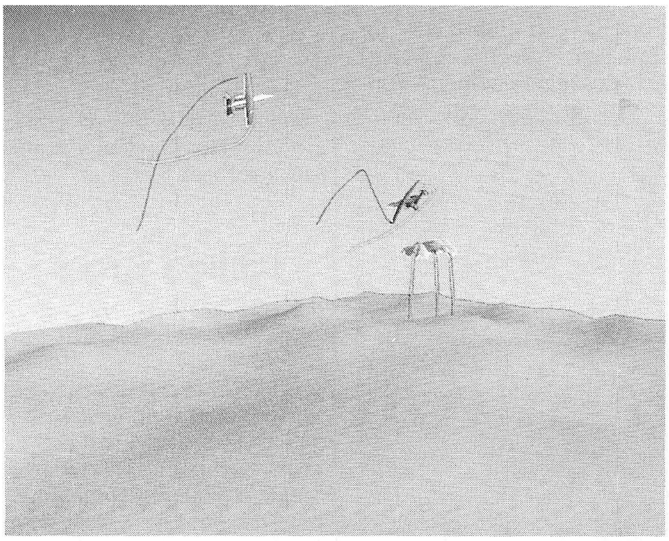
Vortex, een fabrikant uit Duitsland, heeft ATonce-Amiga aangekondigd. ATonce is een AT-emulator voor de Amiga 500 waardoor de Amiga verandert in een MS-DOS computer met een 80286 processor met een snelheid van 16 bits/8 MHz. ATonce loopt vanaf DOS 2.11 en wordt als een kaart op de 68000-sockel geprikt. Daarboven op komt de processor dan weer te zitten.
ATonce ondersteunt harddisks en maakt van alle aangesloten drives, MS-DOS drives (3,5"/720KB, 5,25"/360KB). DOS wordt direct van de harddisk of de flop geboot.
ATonce draait op een Amiga met minimaal 1 MB. De emulator geeft dan 640 KB onder MS-DOS ter beschikking. Verder RAM-geheugen kan als extended/expanded geheugen worden aangesproken.
De ATonce-Amiga emuleert CGA, Hercules, Toshiba T3100 en Olivetti G0317. De Amiga muis verandert in een Microsoft muis en de eventuele ingebouwde klok in een AT-klok. Ten slotte wordt de seriele poort omgetovert tot COM1 en de parallelle tot LPT1.
De prijs van de ATonce zal onder de 500 Mark liggen, zonder MS-DOS overigens. De emulator is vanaf 1 oktober leverbaar.

Lattice Cross Compiler V 5.0

Lattice Inc. heeft een nieuwe versie aangekondigd van de MS-DOS naat AmigaDOS C cross compiler. De nieuwe versie is volledig compatible met de huidige versie van Lattice AmigaDOS C Development Systeem.
De nieuwe versie wordt compleet met dit systeem geleverd. Zo is het mogelijk Amiga programma's te maken vanaf een MS-DOS systeem.
Geregistreerde gebruikers kunnen updaten tegen $ 350. De prijs van het pakket zelf ligt op $ 750.
Inl: (09-1)7089161600 (tel), (09-1) 7089161190 (fax)

Virtual Reality:

Van spel naar driedimensionale verbeelding



CDTV eerste stap naar totale integratie video.

Wie kent niet de kleurrijke 'adventure' actiegames of de zeer complexe tekstadventures, waar men uren mee bezig kan zijn, wegdromend in een wereld vol ridders, helden en gevaren. Maar stel je eens voor, dat je die avonturen ook kunt beleven in een echte driedimensionale omgeving, waarbij je geen joystick meer nodig hebt, maar de bewegingen van de hoofd, je voeten en je vuisten worden overgebracht in de computer.

Met de computer op reis, naar werelden die net zo echt of onecht kunnen zijn als men wil. Met een speciale helm en speciale natuurgetrouwe controls kan men zo binnenstappen in een omgeving, die net echt lijkt maar toch alleen in het geheugen van de computer 'en de ervaring van de reiziger' bestaat. Virtual Reality wordt gezien als de grote doorbraak in interactief computergebruik.

Amiga

In het hele spel van de Virtual Reality (VR) speelt de keuze van de hardware, het zogenaamde 'platform' een grote rol. Er zijn krachtige machines nodig, die veel en snel kunnen werken met vooral video-informatie. De Amiga blijkt, naast de professionele workstations die door grote bedrijven worden gebruikt, als de 'budget'-machine voor deze toepassing de logische keus.
Jack Tramiel zal er nog spijt van krijgen, dat hij destijds Amiga voor Commodore in de waht sleepte. Hij heeft nog wel geprobeerd om het bedrijf via juridische machinaties naar Atari over te halen, maar dat is niet gelukt. Hij wist toen al, dat er in de Amiga aanpak een paar hele leuke dingen zaten. In gesprekken met Jack heeft hij wel eens verteld, dat de hele consumentenmarkt en de trend naar meer pixels en meer kleuren eigenlijk draaide om de RAM prijs. Omdat daar de laatste jaren wat problemen mee waren, kon hij ook voor de Atari ST niet snel genoeg en goedkoop genoeg de MegaBytes de huiskamer inschuiven. Meer RAM betekent snellere verwerking en vooral meer complexe software.
De Amiga heeft qua ontwerp altijd klaar gestaan voor veel RAM en dan met name voor gebruik in de video-processing. Atari had via de MIDI een betere ingang naar de audio toepassingen, maar vanaf het begin was duidelijk dat de Amiga technisch gezien eigenlijk de beste kaarten had. Zeker met de 68040 processor, waarvan sprake is (er zijn al kaarten van o.a. Supra voor 1200 $) en de interface naar grote opslag zoals CD-ROM en wat later CD-I hoeft een Amiga niet onder te doen voor professionele kleurenwerkstations. Vooral de prijs van de software en het relatief gemakkelijk koppelen en mengen met video- beelden gaat de Amiga voor VR een stevige impuls geven.

CDTV

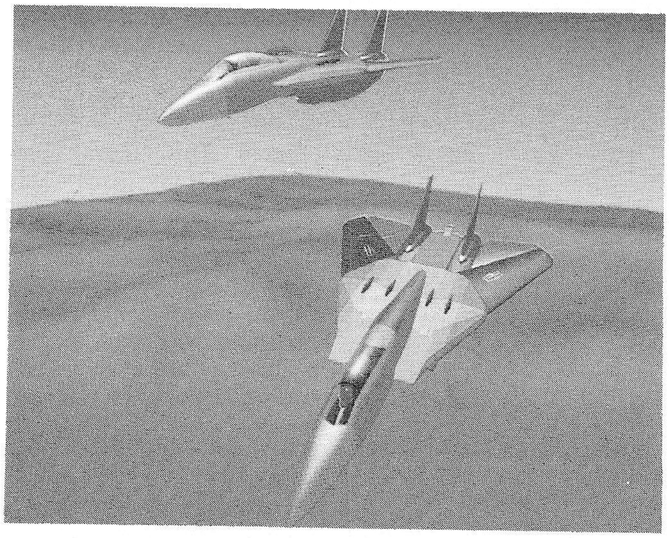
Nolan Bushnell, die nu druk bezig is met de CDTV, heeft wel degelijk het oog op de VR toepassingen. Hij was de stichter van Atari, maar werkt nu weer in West Chester samen met Gail Wellington, en dat is een echte Commodore oudgediende, aan dergelijke multi-media projecten. Dat men verder wel dan de standaard toetsenbord aanpak blijkt eruit, dat de bediening van de Amiga CDTV (Commodore Dynamic Total Vision) met ingebouwde CD-ROM direct via een infrarood besturing gaat. men mikt op de echte 'electronische boeken', maar daarna ook op de interactieve video en de interactieve virtual reality. En zoals altijd bij Commodore, voor een groot publiek tegen redelijke prijs.
Maar laten we eerst even terugkomen op wat Virtual Reality nu eigenlijk inhoudt.

Veelzijdiger en vrijer

Er begint langzaam verandering te komen in de vrij eenzijdige toepassing van de computer voor alfanumerie-

ke gegevensverwerking. Hier en daar komen toepassingen naar voren, die buiten het gebaande automatiseringspad vallen. Vreemde zaken, zoals de psycho-actieve software, maar ook toenemende integratie van telecommunicatie en computerkracht.

Virtual Reality ofwel Cyberspace is zo'n opwindende blik in wat we de volgende eeuw kunnen verwachten.



Dit is het gebied waar de creatiefste computer talenten zich op dit moment in vastbijten. We noemen Ted Nelson, Jaron Lanier, Erich Gullichsen, maar ook Timothy Leary is actief met Virtual Reality, dat hij al heeft aangeduid als LSD per computer.

Stereoscopische recreatie

Lang leve de computer als stimulerende vriend en metgezel, weg met de duffe cijferpeuteraars. Het ziet er naar uit, dat we langzaam gaan ontdekken dat een computer nog een heleboel meer is en kan dan alleen maar databank en boekhoudmachine. We gaan een nieuwe tijd in en ontdekken dat achter het beeldscherm misschien wel een hele andere wereld ligt, een waarbij de gewone regels van de realiteit niet meer opgaan. Want normaal gesproken kunnen we niet naar willekeur omhoog en omlaag, springen en vliegen, of gewoon een heel andere wereld instappen. We kunnen geen nieuwe 'entities' maken en daar mee communiceren, wanneer we eens door de sleur van de dagelijkse realiteit willen heenbreken. Wanneer we de computer gebruiken om wat 'Virtual Reality' (VR) heet te simuleren, op een scherm of op een speciale bril met mini-schermpjes voor driedimensionale effecten, kunnen we de grenzen van het normale overschrijden.

Overal

De toepassingen van VR zijn alleen maar beperkt door onze verbeeldingskracht. Dr. William Bricken maakte een kleine opsomming, waarin Virtual Sports, Air Traffic Control, Virtuele ledematen voor gehandicapten, de Virtuele psycho-analist, de Virtual Decorator om het eigen huis met andere meubels en kleuren te bekijken, Virtuele concerten, en natuurlijk de Virtuele School, maar er is nog veel meer te bedenken.
Met de juiste apparatuur is het al mogelijk om middels de computer als wandelaar in een nog niet bestaand ontwerp van een gebouw rond te dwalen. Men kan dan rondkijken, de invloed van afmetingen en kleuren van het ontwerp proeven alsof het gebouw er al stond. AutoDesk, de fabrikant van AutoCad, heeft in onder meer haar Cyberia project al dergelijke ontwikkelingen gerealiseerd. De toepassingen van VR liggen op heel praktisch gebied, miltaire toepassingen en ruimtevaart, maar gaan ook zo ver als Teledildonics, sex op afstand.

Reizen in stilstand

Een goed voorbeeld van 'Virtual Reality' (VR) is het rondreizen in bijvoorbeeld vreemde landen. Via een speciale helm met een stereo- videodisplay (2 schermpjes) met een ruimtelijk effect kan men gaan rondkijken in een 3-D omgeving, bijvoorbeeld een exotisch land. Zonder uit de stoel te komen kan men daar, al naar gelang het programma het toelaat, interactief in rondkijken. Via een speciale sensor worden de hoofdbewegingen namelijk vertaald in bewegingen in die wereld en kan men verder via een joystick de richting van de beweging aangeven.
Er wordt nu al gesprokene over 'spacemakers', mensen die dergelijke werelden en reizen opzetten. Een nieuw beroep, dat alleen met de computer bestaat, nu eens geen electronische vervanger van bestaand werk.

Meer dan computerspelletjes

De meeste gebruikers van een PC hebben wel eens met een simulatieprogramma zoals Flight-simulator gespeeld. Daarbij zit met als het ware (virtueel) in de stoel van de piloot en vliegt door een wereld met nagebootste vliegvelden, andere vliegtuigen en contacten met verkeersleiders etc. Bij dergelijke simulators is er sprake van gesimuleerde bewegingen in een abstracte ruimte. Dat wordt in de spelletjessfeer nog duidelijker, ook de ruimteschip shoot-out varianten vallen in deze categorie.
Een andere invalshoek vormen de Adventures. In spelen als Powermonger of Populous zien we al, dat de speler heer en meester is over een bedachte wereld, desnoods met wat medespelers. Maar met VR kan hij daar dan ook nog echt in rondreizen.

Ervaringen

Ik werd zelf voor het eerst blootgesteld aan VR bij Tim Leary thuis in de heuvels van Hollywood, eind 1989. Erich Gullichsen demonstreerde daar aan mij en John Barlow zijn Sense8 installatie. Dat werkte op basis van een Amiga (toen nog 2000) en met een speciale ronde joystick/3-D stuurknuppel en stereo-helm. De VR wereld, waar ik in mocht kijken was een van de eerste probeersels op dit systeem. Het rondreizen in een soort 3-D spelomgeving met behulp van de stereo-helm was niet zo verbluffend. In veel video-spelletjes en flight-simulators heeft men ongeveer dezelfde macht over de omgeving. Natuurlijk is het vreemd om bij het draaien van het hoofd het beeld te zien veranderen en meegaan, maar dat went snel. De aha-erlebnis kwam, toen ik na een kwartiertje weer de helm afzette en opstond, en toevallig voor een abstract schilderij stond. Ik had toen even het gevoel, dat dat ook driedimensionaal was en er er zo in kon lopen, dat het schilderij ook zo'n VR wereld was. Mijn bewustzijn en perceptie waren duidelijk even op het 'verkeerde' of juist goede been gezet. Op dat moment besefte ik, waarom men dit LSD per computer noemt en waarom de hallucinatie-freaks uit de 60-er jaren dit zo'n te gekke technologie vonden.

Illusies

Bij 'Virtual Reality' speelt illusie een grote rol, de mens laat zich relatief snel overtuigen van de 'werkelijkheid'

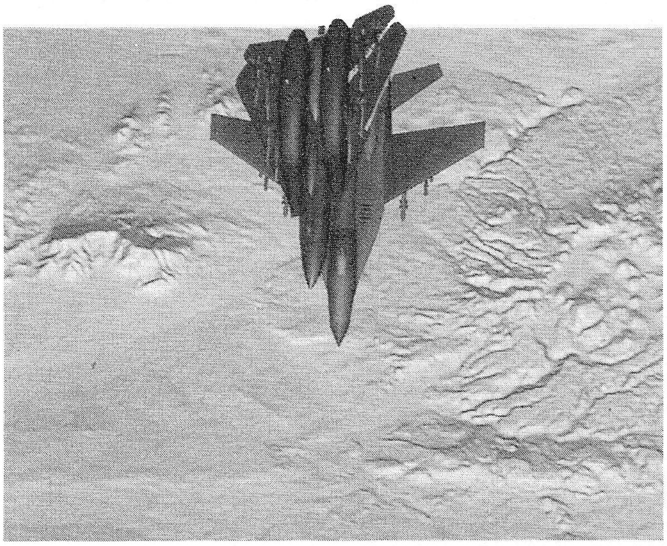
van zo'n niet- of half-bestaande omgeving. Randall Walser van AutoDesk zegt: 'Cyberspace is eigenlijk een soort magie, het creëren van illusies'. De meest verbreide vorm van praktische Virtual Reality is dan ook het 'Adventure-game', waar de speler door een voorbedachte wereld trekt. Daarvoor gebruikt men vaak magische symboliek. De speler identificeert zich al snel met de held uit zo'n spel, en raakt verslingerd aan zijn 'King's Quest'. Maar stel eens voor, dat zo'n spel zich afspeelt in een driedimensionale omgeving, die men via een stereoscopische projectie ook 3-D kan zien, waar men dingen kan voelen en pakken via een 'Dataglove', men andere spelers kan ontmoeten en daarmee in real-time kan vechten of wat dan ook? En nog een stapje verder, wanneer men die imaginaire omgeving zelf kan bedenken en vorm geven.

Jeffrey Shaw

De Amsterdamse kunstenaar Jeffrey Shaw, afkomstig uit Australie, en zijn collega Dirk Groeneveld maken computerkunst waarbij 'Virtual Reality' een grote rol speelt. Ze proberen een 'environment' te maken, waarvan de kijker/gebruiker actief deel uit maakt. 'De leesbare stad' (The Legible City) is een interactief project dat o.a. in de Amsterdamse Balie aan het publiek is getoond, en waarmee men door een abstracte ruimte kan reizen.

Leren omgaan met ruimte

Er zijn situaties en 'werelden', die men in het echt niet zomaar kan betreden. In de ruimtevaart, maar ook bij het analyseren van de constructie van een atoomreactor kan een 'cybernaut' nu rustig wennen aan een vreemde omgeving. Voor trainingsdoeleinden, maar ook als gewoon amusement. Men bestuurt gewoon een 'puppet', en kan dan vanuit de stoel op avontuur gaan. In VR kan men vanuit de stoel een tripje naar de amazone maken, maar ook gaan basketballen in een wereld, waar de zwaartekracht niet lineair is. Juist doordat men moeiteloos van gezichtspunt en positie kan verwisselene ontstaat al snel een gevoel van vrijheid, van transcendentie, dat een beetje lijkt op wat men beleeft bij een 'trip' onder invloed van bijvoorbeeld LSD. Die verbinding werd dan ook door velen gelegd, de onderzoekers op dit gebied zijn daar niet blij mee, men vreest beperkingen door de overheid. De 'Drug per computer' heeft echter beslist onwerkelijke en fascinerende trekjes, zeker wanneer men in wat 'vreemdere' werelden gaat rondreizen. Gibson sprak al over '*consensual hallucination*' in de Scifi roman Neuromancer. Zo kan men in een VR allerlei objecten een soort identiteit geven en op die manier een rollenspel gaan uitvoeren. men hoeft ook niet alleen in een wereld te vertoeven, maar kan die delen met andere Cybernauts. Dat lijkt toekomstmuziek, maar in 'Virtual Reality for Two' aanpak van pionier op dit gebied Jeron Lanier is er sprake van een virtuele ontmoeting in een virtuele ruimte tussen twee participanten, die ieder alleen direct contact hebben met hun computer.



Cyperpunk

Er zijn een aantal woorden met ongeveer dezelfde betekenis als Virtual Reality (VR). Seymour Papert (logo) gebruikte 'microworlds', Ted Nelson 'Virtuality'. De kreet Cyberspace hangt samen met Cybernetica (Norbert Wiener) en met Cyberpunk, en dat is weer een speciale categorie Science-Fiction literatuur uit de VS.
Bill Gibson was met Neuromancer duidelijk de trendsetter op dat gebied, hij was laatst nog in Nederland op het Sound & Image evenement in het Haagsche Filmhuis. In de VS is Cyberpunk een grote rage onder met name de hi-tech afficionado's, de hackers en de computergebruikers. Bill Gibson schilderde de Cyberspace als een nieuw medium, een abstracte wereld bestaande uit electronische data, beelden en verbindingen, direct toegankelijk voor iedere gebruiker (operator).

Historie

Het is moeilijk om aan te geven, wie nu precies de vader van Virtual Reality is. Morton L. Heilig heeft de oudste octrooien op dit gebied, die dateren al van 1957 (stereoscopische videobril) en 1961 (Sensorama Simulator).
Een van de eersten, die de computer zag als fundamenteel werktuig voor de transformatie van het menselijk denken en cratieve activiteit was de Amerikaan Vannevar Bush. In 'As we may think' uit 1945 wees hij erop, dat de mens betere middelen nodig had om door de steeds wassende zee gegevens heen te komen dan de klassieke wetenschappelijk aanpak met publikaties, boeken en bibliotheken. Hij voorzag een (toen heel futuristisch) Memex apparaat, dat als persoonlijke databank zou gaan fungeren, en waarin teksten, plaatjes, notities en indexen worden opgeslagen. Een mechanisme voor snel doorlopen en doorbladeren van (microfilm) teksten en indexen, linken van tekstpassages door een trail, Bush beschreef hier al delen van wat we later bij Hypertext tegenkomen. Zijn visie op een multimedia aanpak, zoals we die in de 90-er jaren realiteit zien worden met o.a. CD-ROM en WORM, bleek profetisch.

AutoDesk

Bij AutoDesk, waar men aan het programma AutoCad voldoende verdient om wat te kunnen experimenteren, startte John Walker het Cyberspace Initiative, dat nu project Cyberia heet. Met in het achterhoofd om een stereoscopische versie van AutoCad te maken, maar men besefte al snel dat 'Cyberspace' veel meer inhield dan een hulpmiddel voor architecten. Cyberia is de afgelopen paar jaar de leerschool geweest voor heel wat VR specialisten, die deels al weer zijn uitgezwermd. Randy Walser is nu de belangrijkste VR man bij AutoDesk.

VPL

Jaron Lanier is de guru van de VR. Hij heeft niet alleen technisch interessante dingen ontwikkeld met zijn bedrijf VPL, maar ook theoretsch zijn visies baanbrekend. VPL heeft de eerste 'Datglove' gemaakt, waarvan een afstammeling als de 'PowerGlove' is uitgebracht voor het Nintendo systeem. Naast de helm is namelijk ook de manuele interactie met de omgeving erg belangrijk voor VR reizen en dus zijn er ondertussen 'Datasuits' en allerlei feedback apparaatjes. Men zou ook kunnen werken met spraak, een keyboard of joystick, maar in veel 'echte' omgevingen is dat wat gekunsteld.
Jaron Lanier is een zonderlinge man, die naast VR ook veel aan muziek doet. Toen ik hem ontmoette in Berkeley tijdens het nieuwjaarsconcert van de Grateful Dead, was hij nogal 'spacy' en niet erg communicatief, maar volgens sommigen leeft Jaron gewoon op een ander vlak. Hij ziet er niet uit, met rasta haar en oude kleren, maar de VR werelden die hij maakt zijn imponerend. Jaron heeft veel publiciteit gekregen, waar hij niet erg blij mee is. Vooral de vergelijking tussen VR en LSD vindt hij onterecht en gevaarlijk.
Jaron Lanier ziet erg veel in VR als een ontmoeteingsplaats. Hij denkt, dat mensen elkaar in de toekomst zullen ontmoeten in zo'n VR en dan daar ook met elkaar kunnen communiceren. VR wordt de favoriete ontmoetingsplaats, niet beperkt door materieële randvoorwaarden, alles kan en alles mag. Al naargelang de sensors en feedback beter worden kan men dan verder gaan dan de datagloves van nu, maar werken met datasuits. Voor de weergave zit het erin, dat men uiteindelijk uit de helm zal groeien en bijvoorbeeld met holografische projectie zal gaan werken. Op die manier komen we weer bij de Cyberspace ideëen van Gibson uit.

Sense8

De goedkope manier om VR te realiseren draait rond beschikbare hardware. Erich Gullichsen en Pat Gelband gingen bij AutoDesk weg om op basis van de Amiga een budget-VR systeem te gaan ontwikkelen. Hun bedrijf Sense8 gebruikt de Amiga 3000, de Dataglove en een goedkope 3-D helm. Hun systeem heeft de codenaam OS/3D. Hun software om zelf VR werelden te maken heet WorldTool.

Auteurs

Er zijn een aantal schrijvers, die zich met name op het Cyberspace gebied begeven. Tim Leary schrijft er over, maar het zijn met name Howard Rheingold en John Barlow, die de in de diverse Amerikaanse bladen het best geinformeerd blijken. Rheingold is de editor van de Whole Earth Review en heeft een Cyberspace nieuwsbrief. John Perry Barlow is o.a. lyricist van de Grateful Dead en een bekende 'Technology Mystic'. Zijn interviews met Jaron Lanier zijn spraakmakend geweest en hij is nu samen met Mitch Kapor (Lotus stichter) een actie begonnen on de tamelijk subversieve acties van de FBI en CIA tegen de hackers in beeld te brengen. Dr. William Bricken is een van de wetenschappers (U. van Washington) die zich met VR bezighouden, maar hij komt ook van AutoDesk af. Myron Kruger schreef in 1972 al het boek 'Virtual Reality'.

Cyberspace display

Er zijn al een aantal produkten voor Cyberspace/VR toepassingen op de markt. Een aantal CAD programma's, met name op Silicon Graphics en andere zware grafische workstations heeft een stereo-3D of 4-D optie.

Ook voor de ST en Amiga zijn er al stereoscopische brillen en ook wat programma's voor 3-D effecten voor zowel de Amiga als de Atari ST op de markt, maar de kwaliteit van die beelden is nog beperkt (Haitex X-Specs 3-D voor Amiga). Men bereikt het 3-D effect door te werken met een LCD bril die afwisselend links en rechts 'opengaat', en dan wordt tegelijk het corresponderende beeldje getoond. Op zich leuk en ook iets wat nog veel verder kan worden ontwikkeld, maar de echte stereobril is toch kwalitatief beter. Virtual Reality Labs uit San Louis Obispo (CA) heeft al diverse programma's voor de Amiga, waaronder Vista, Fractal Flight en Distant Suns.
Cyberspace/Patient Communications uit Norcross, Georgia USA heeft een mini-display ontwikkeld, dat men heel dicht voor de ogen kan gebruiken. Daarmee kan een EGA beeld worden afgebeeld, zonder dat men vastzit aan de normale zware beeldbuis. Voor gehandicapten en allerlei medische en industriele toepassingen is dit een uitkomst.

Luc Sala

Boeken: Godel, Escher, Bach door D. Hofstadter '79
Human Computer Interaction : Preece/Keller '90 Prentice Hall
Artikelen:
Cyberspace J. Marshall Electric Word dec. 89.
Cyberspace Mondo 2000 sept '89
Synthetic Generated realities, Rheingold Whole Earth summer '90
C 1999 Dr. W. Bricken Mondo 2000 jun '90
Being in nothingness J. Barlow Mondo 2000, jun '90
Nieuwsbrief over Cyberspace: Howard Rheingold.

Timothy Leary: Virtual Reality

Op 14 september komt Timothy Leary in Paradiso in Amsterdam om een lezing te geven over het onderwerp Cyberspace en LSD per computer.

Hij is erg bekend geworden als de Harvard professor die in de zestiger jaren LSD propageerde. In de tachtiger jaren heeft hij zich ook op de computer toegelegd en onder meer psychologische spelsoftware (Mind-Mirror) gemaakt. Timothy Leary is in de VS enerzijds een omstreden figuur vanwege zijn denkbeelden, aan de andere kant heeft hij zich ontwikkeld tot een echte 'new-age' en 'new-edge' voortrekker. Hij is al 72, maar is nog zeer actief en schrijft nog reeksen artikelen en boeken. Vanaf het begin heeft hij de potentie van VR ingezien en actief meegewerkt aan de ontwikkeling. Naast Tim komt ook de echte VR ontwikkelaar Gullichsen spreken op deze avond. Omdat we verwachtten, dat vanuit de computerwereld grote belangstelling zou ontstaan voor het bijwonen van de lezingen van Timothy Leary, hebben we een aantal kaarten weten te reserveren voor zijn presentatie op 14 september in het Amsterdams Paradiso. Bel daarvoor 020-827678 of fax 260929.

VECTORTRACE

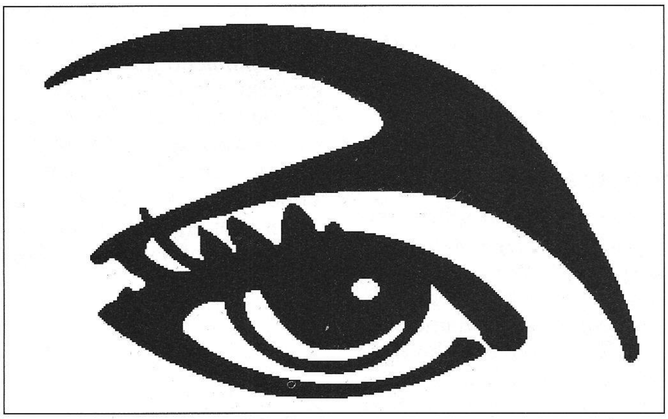
Punten worden lijnen

Iemand die wel eens IFF-plaatjes in programma's laadt, krijgt te maken met het begrip pixels en vector, zonder dat de persoon in kwestie het zelf weet. Plaatjes zijn namelijk uit puntjes opgebouwd. De opbouw is in zekere zin vergelijkbaar met een afdruk van letters afgedrukt op een matrixprinters. Voor bepaalde plaatjes geldt ditzelfde: aan de kanten vallen de kartelingen op.

Iedereen die wel eens IFF-plaatjes op een printer hebben afgedrukt, heeft die kartelingen wel eens gezien. De pixels zijn hier de oorzaak van. Programma's als Turbo Print (Professional) kunnen dit enigszins opheffen doordat zij de mogelijkheid bezitten de randen kunstmatig gladder te maken. Ook WorkBench 1.3 kwam hierin enigszins tegemoet. Maar iemand die diezelfde plaatjes in een DTP-programma wil gebruiken, komt al snel in de problemen.

DeskTop Publishing

DTP-pakketten hebben vaak eigen printer-drivers, en kunnen daarom geen gebruik maken van bijvoorbeeld Turbo Print. Verder komt het vaak voor dat een DTP-er de IFF-plaatjes wil vergroten, omdat het dan beter in een bepaalde stijl past. Wat dan extra groot naar voren komt, zijn weer de randen van de plaatjes. De kartels worden met het vergroten van het plaatje meevergroot. En hoe



Een normaal IFF-plaatje voordat het met VectorTrace bewerkt werd

de afdruk er uiteindelijk uit gaat zien, laat zich raden.

Toverwoord

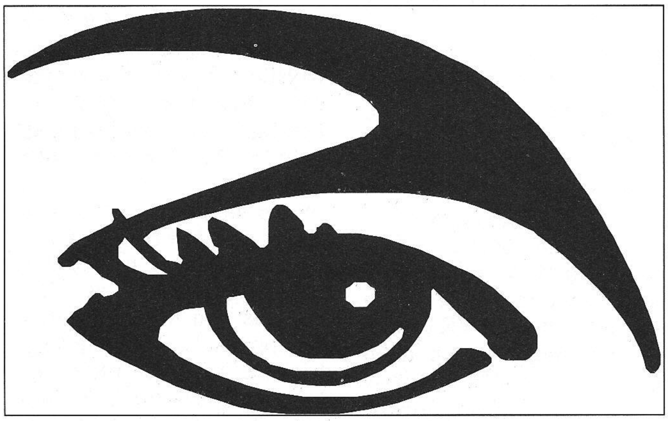
Op dit probleem heeft Gold Vision uit Berlijn iets gevonden: VectorTrace 1.1. Dit programma converteert IFF-plaatjes naar verschillende formaten die door andere programma's, bijvoorbeeld DTP, weer kunnen worden ingelezen. De formaten die VectorTrace ondersteunt zijn: Aegis Draw, Encapsulated PostScript, Professional Draw-Clip en Video 3D-Format.

Opstarten

VectorTrace kan vanaf de CLI worden gestart met een aantal mogelijkheden:

- zonder toevoeging, het programma wordt vanuit een eigen interlace-scherm gestart;
- -n, het programma wordt in de non-interlace modus gestart in dezelfde grootte als het Work-Bench-scherm;
- -s, VectorTrace start in een eigen scherm met de standaardgrootte 640x512 (PAL) of 640x400 (NSTC);
- -w, het programma start ook nu in een eigen scherm en dan op de WorkBench. De laatste methode kost het minste geheugen en is daarom met name bedoeld voor de gebruikers met 512 KB geheugen.

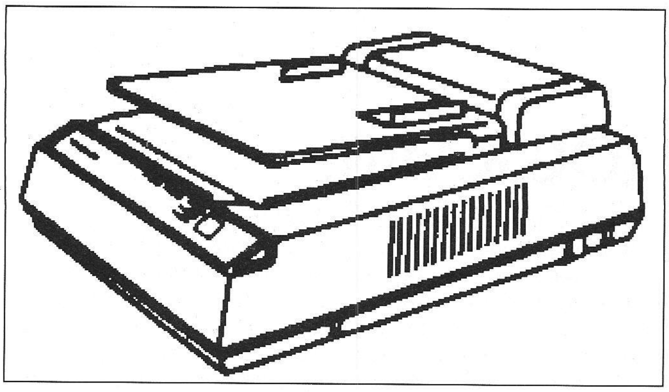
VectorTrace kan overigens door middel een icon worden gestart. De opties kunnen desgewenst ook in het Info-venster worden meegegeven. Om iets met het programma te kunnen doen, moet er eerst een IFF-plaatje worden ingelezen. Wanneer dat gebeurt is, moet het worden aangeklikt. Dit is nodig voor het bewerken van het plaatje.



Het resultaat na de bewerking in VectorTrace.

Bewerken

Het bewerken van het plaatje doet VectorTrace helemaal zelf als de opdracht daartoe gegeven. Afhankelijk van de grootte hiervan, duurt tot maximaal tien minuten op een normale Amiga (68000). Als dat gebeurt is wordt het plaatje in het gewenste for-



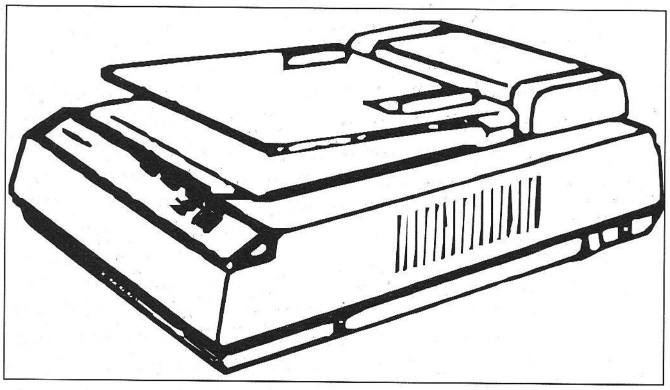
Voor het inladen

Om het meeste resultaat te krijgen, schreven we de plaatjes ook nog eens weg in PostScript-formaat.
De hoge resolutie zorgt ervoor dat er eigenlijk helemaal niets meer van de puntjes is weer te zien. Nadeel is dat je hier een PostScript-compatible printer voor nodig hebt, die niet altijd voorhanden is.

Maar het resultaat maakt in ieder geval duidelijk dat degene die IFF-plaatjes voor publicaties wil gebruiken in een DTP-programma, met VectorTrace goed af is.

Inl: 3Gitaal, 020-970035

maat naar de diskette weggescheven.
Het inladen in andere programma's hebben we geprobeert en met succes!. Ook PostScript-files konden zonder problemen worden uitgeprint.
De illustraties laten trouwens zien waartoe VectorTrace in staat is. Deze zijn als IFF ingeladen. Daarna werden ze geinventeerd. Vervolgens lieten we VectorTrace het plaatje herkennen en helemaal bewerken. Het formaat in Aegis Draw vormde geen enkel probleem. Ook het inlezen in het DTP-programma Pagestream ging zonder problemen.



Na het bewerken

AMIGA BIBLIOGRAFIE

Het afgelopen jaar is er vrij veel aandacht geweest aan het diepere Amiga-wezen. Dat hier een welwillende leesmarkt voor is, mag blijken uit de nog steeds groeiende stroom Amiga-literatuur. Uit deze stroom hebben we vier boeken gehaald welke we eens nader onder de loep nemen.

De Amiga literatuur laat zich grofweg scheiden in twee delen. Het eerste deel richt zich op de gebruiker van de eindprogramma's, die mensen die geen behoefte hebben om zich bezig te houden met alle diepere zaken van hun computer. Het tweede deel richt zich op die groep die juist meer wil weten van hun (vaak) trouwe bureaudier. Deze boeken gaan dieper in op de zaken.

Amiga Intern 2.0

Dit boek, dat door Data Becker wordt uitgebracht en door Bruna uitgevers B.V. in Nederland wordt gedistribueerd, bevat een schat aan gegevens voor de doorgewinterde programmeur. Het boek begint met een aantal afspraken om tot netjes programmeren te komen, waarna echt de gaspedaal tot de bodem wordt doorgeduwd. Zaken die bij de Amiga als toch redelijk pittig bekend staan, zoals het programmeren van de devices, worden overzichtelijk en helder uitgelegd. Rijk voorzien van, helaas nog steeds niet altijd goed werkende, voorbeelden, wordt de tekst alras duidelijk aan de nog niet volleerde programmeur.
Wanneer u geïnterresseerd bent in de verschillende IFF-formaten, ook hieraan is in het boek een apart hoofdstuk gewijd. Het vijfde hoofdstuk is de droom van menig doorgewinterde programmeur. Alle WorkBench 1.3-functies, onderling van elkaar gescheiden, worden behandeld. Wilt u een functie schrijven? Zoek eerst maar eens in dit boek, voordat u zelf aan de slag gaat! Alsof dit nog niet genoeg is, bevat het het laatste hoofdstuk als toegift een complete verhandeling over het zelf programmeren van printerdrivers. Altijd al een secret wish van yours truly geweest. Tot de laatste bladzijde blijven de schrijvers sterk doorgaan, wilde u altijd al weten hoe u uw eigen fonts moest programmeren? Dit boek vertelt het.
In het kort gezegd, een kei van een boek waaraan u nog veel programmeer-plezier zult beleven.

Amiga Intern 2.0
Bleek, Jennrich & Schulz
Data Becker, A.W Bruna uitgevers B.V.
ISBN 90 229 3683 X
f 89,90

Het grote Amiga C boek

Dit boek laat zich bijna omschrijven als de alternatieve handleiding van de Aztek C v3.6 compiler. Het eerste hoofdstuk is in z'n geheel gewijd aan de beschrijving en werking van de compiler, de linker en de assembler. Vanaf hoofdstuk 3 begint het verhaal pas echt. Intuition wordt in al z'n gedaanten onthuld. Hoe programmeert u requesters, menu's, gadgets, windows, alerts, screens, het consoledevice of de IDCMP-facility van de Amiga, al dit wordt behandeld in dit hoofdstuk. Bent u geïnteresseerd in de altijd mysterieuze Guru foutmeldingen? Op bladzijden 192 t/m 195 worden de foutnummers stuk voor stuk ten tonele geleid.
Vanaf hoofdstuk vier begint het tweede deel van het boek. De auteurs leggen in dit hoofdstuk uit hoe een editor in C geprogrammeerd kan worden. De tekst wordt continu ondersteund door programmadelen, die eenmaal in elkaar gezet, een echt werkende editor moeten vormen.
Voor beginnende Intuition-programmeurs een steun en toeverlaat. Voor gevorderden en boek om steeds weer op terug te vallen.

Het grote Amiga C-boek
Bleek, Jennrich & Schulz
Data Becker, A.W. Bruna uitgevers B.V.
ISBN 90 229 3483 7

Amiga programmer's handbook vol.2

Dit Engelstalige boek laat zich bijzonder moeilijk lezen. Dit niet zozeer door de Engelse taal, maar door de weinige voorbeelden. Desondanks bevat het boek een schat aan gegevens, die pas na een paar maanden door het boek worstelen eruit kunnen worden gehaald.
Het boek beschrijft op een duidelijk Amerikaanse manier de bediening van de devices. De werking van de devices wordt wat naar de achtergrond gedrukt, men is meer bezig met het proberen te verduidelijken van het programmeren van de devices. Dat deze materie vrij moeilijk is, mag blijken uit het feit dat ondergetekenden het boek al enige tijd bezitten maar vrij veel moeite hebben gehad met het lezen van het boek.
De maker is er helaas niet geheel in geslaagd de materie duidelijk te presenteren. Door het gebrek aan voorbeelden kunt u het gevaar lopen in een moeras van woorden te verzinken. Echter indien men ook de Amiga Intern 2.0 bezit, dan is dit boek een unieke aanvulling op de gegevens in de Intern.

Amiga programmer's handbook
Eugene P. Mortimore
SYBEX Inc.
ISBN 0 895888 384 8
f 76,=

Inside the Amiga with C

Dit toch al weer enige jaren oude boek blijft het nog steeds goed doen bij ondergetekenden. De schrijver tracht alle aspecten van de Amiga te beschrijven. Dit met vele voorbeelden waardoor u al snel een werkend programma zal op kunnen zetten. Alle Amiga-zaken worden in dit boek behandeld, wat helaas resulteert in een te geringe diepgang. Kort uit de inhoud, hoe programmeert men Intuition in al z'n specifieke gedaanten, hoe kan men van de multitasking aspecten van de Amiga profiteren, sprite-animaties, het programmeren van geluid en spraak en de disk i/o.
Kortweg, niet te diep maar toch net diep genoeg om een idee te geven van de mogelijkheden en aspecten van de Amiga.

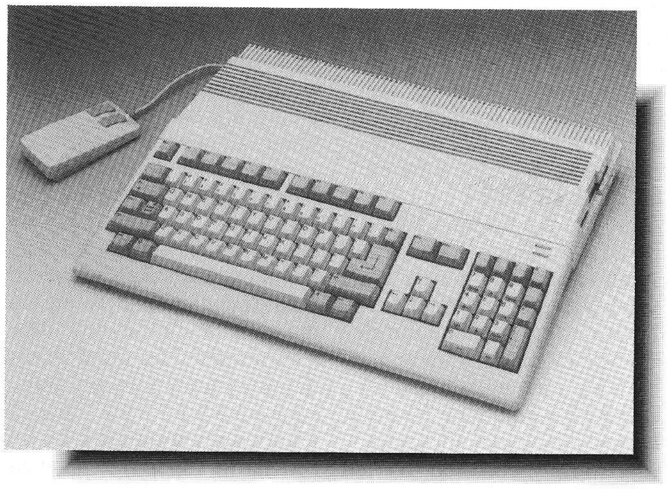
Inside the Amiga with C
J.T. Berry
Howard W.Sams & Company
ISBN 0 672 22468 2

Johan & Johan

Deze boeken zijn leverbaar bij Computer Collectief

AMIGA STARTERS

Deze keer een uiteenzetting van de verschillende poorten op de Amiga.



Change Serial-gadget

Als u gebruik maakt van dit Gadget dan zult u een nieuwe window te zien krijgen. In dit window kunt u dan allerlei veranderingen aanbrengen met betrekking tot de seriële poort. Wat houdt de seriële poort eigenlijk in?
Zoals u weet (moet weten) zitten er aan de achterkant van de Amiga een aantal aansluitingen. Op deze aansluitingen kunt u hardware-aansluiten als een DiskDrive, Harddisk of modem. In uw handleiding staan alle aansluitingen vermeld en welke waarvoor dient. Toch zullen we ze één voor één de revue laten passeren, voordat we weer terug komen bij de Change Serial-Gadget.
Eerst nog een algemene waarschuwing. Als u de computer uitschakelt doe dit dan doormiddel van de aan/uitschakelaar op de voeding. Trek alstublieft niet de stekker uit het stopcontact als aan- of uitschakelaar, aangezien uw computer of voeding daar niet van gediend zijn. Dit is ons namelijk ook een keer noodlotig geworden. Als uw computer dan nog in de garantie zit is er niet veel aan de hand maar alles bij elkaar duurt het toch weer een paar weken voordat de computer weer uit zijn coma teruggekeerd is. En wat is er nou erger dan uw AMIGA een paar weken te missen. Slapeloze nachten zijn het gevolg, families gaan er kapot aan, de gevolgen zijn gewoon niet te overzien.
Waarschuwing twee moeten we ook niet vergeten. Bij het aansluiten van een randapparaat is het verstandig uw computer uit te zetten. Doet u dit niet, dan kan dat leiden tot beschadiging van de randapparatuur of van uw computer.

Muis/Joystick konnektor

We beginnen, van achter af gezien, aan de linker kant. Hier zult u twee gelijksoortige aansluitingen vinden. De eerste is bedoelt voor aansluiting van de muis.
De tweede zal meestal gebruikt worden voor de aansluiting van een joystick. Wat nu te doen als u een spelletje hebt waarbij u twee joysticks nodig hebt? Twee keuzes:

1. Haal de muis uit de eerste poort en sluit daar uw tweede joystick op aan;
2. Koop een kastje met een schakelaar erop en u kunt de muis en joystick tegelijk op poort één aansluiten. Het enige wat u dan te doen heeft, is de schakelaar om te zetten.

Een ander voordeel is dat u uw poort op deze manier niet zult molesteren, dit zijn immers nog steeds behoorlijk 'expensive jokes'.

Audio

Hier zult u twee, van gelijke grootte, lijkende konnektoren vinden. Op deze connectoren vinden we het linker- en rechter geluidskanaal van de Amiga. Stereo, dus zodoende vindt u twee connectoren een voor linker- en de ander voor het rechter kanaal. U kunt uw computer geluid via de monitor of T.V. ten gehore laten brengen maar ook is het mogelijk dat te doen via een stereo-toren. Op een monitor of T.V. gaat dat vaak via een Y-kabel. Eén voor beeld weergave en de andere voor het geluid. Als u een euro-connector aansluiting heeft voor uw monitor dan zit het geluid bij deze plug bijgesloten en hoeft u dus niet meer apart uw geluid op de monitor aan te sluiten. Ook kunt u zoals eerder vermeld via de connectoren in de AMIGA de computer aansluiten op een stereo installatie. Dit doet

u dan op de Aux of Auxiliary uitgang van uw toren.

Diskdrive

Zoals het woord al zegt kunt u hier uw tweede drive op aansluiten. De mogelijkheden zijn of een 3,5 inch diskdrive, dat is toch de meest gebruikelijke of een 5 1/4 inch diskdrive.
Deze laatste wordt vaak gekozen, omdat deze schijfjes veel en veel goedkoper zijn dan de 3,5 inch schijfjes. Een andere reden is, de mogelijkheid om MS-DOS software uit te kunnen lezen.
Vooral nu het 'KCS POWER PC BOARD' op de markt is gekomen (zie de testen die zullen verschijnen of al verschenen zijn) zal het gebruik van MS-DOS software op de AMIGA toenemen, immers de software emulator 'Transformer' is ook niet alles, nietwaar? (13% van een IBM-PC, niet bepaald topsnelheid)
Ook de 3,5 inch drive kent z'n voordelen. De meeste commerciële software is op 3,5 inch uitgebracht. U zult dus altijd één 3,5 inch drive nodig hebben.

Seriële

Deze aansluiting wordt meestal gebruikt voor de aansluiting van een modem of een printer. Een modem is een apparaat waarmee men in contact kan komen met een andere computer met behulp van de diensten van de PTT.
Wat doet nu het modem? Eigenlijk heel simpel, de computer werkt met digitale signalen 1 en 0. Deze kunnen niet direct via de telefoonlijn verstuurd worden daar hier gebruik gemaakt wordt van analoge signalen (zoals stemgeluiden, muziek etc). U raadt het al, de modem zet de digitale signalen om in analoge bij verzenden en bij ontvangst zet hij ze weer om van analoog naar digitaal. Vandaar de naam MOdulator-DEModulator. Met een modem kunt u toegang krijgen tot Viditel, Bulletin Boards van bijvoorbeeld AMIGA-gebruikers, informatie opvragen bij instellingen, scholen en andere diensten.

Parallel

Deze connector wordt meestal gebruikt voor de aansluiting van een printer of plotter. De nu volgende typen printers kunt u op deze connector aansluiten. Margriet-wiel printer, deze lijkt op het schrijfmachine type. Het werkt namelijk met een draaischijf waarop de letters staan afgebeeld. Voordelen zijn de hoge letterkwaliteit, nadelen zijn veel lawaai, zeer langzaam, niet grafisch en maar 1 lettertype is tezelfdertijd mogelijk. Dan heb je de matrixprinter deze is op dit moment de meest gebruikte thuisprinter. Dat komt doordat de printers niet duur zijn, een redelijk tot goede letterkwaliteit bieden en in een aardig rap tempo tekst en grafische afbeeldingen kunnen laten zien. Een groot nadeel is vaak de grote hoeveelheid lawaai die geproduceerd wordt. Dat komt omdat de matrixprinter met naaldjes de letters afdrukt op het papier. Dit veroorzaakt over het algemeen een aardige hoeveelheid decibels.
Een oudje die het heeft moeten opgeven ten opzichte van de matrixprinter is de thermische printer. Deze werkt ook met naaldjes die op het papier hameren alleen gaat de afdruk niet met een inktlint maar doormiddel van warme naaldjes die op een speciaal soort papier de afdruk zichtbaar maakt. Dit was een duur soort papier dus het was niet echt rendabel. Als laatste en beste type printer komen we bij de laser printer. Deze printer drukt een hele pagina in een keer af. Voordelen zijn, de goede afdrukkwaliteit (ook grafisch) en het weinige geluidsoverlast. Groot nadeel is de prijs die u voor een laser printer moet betalen.
Voor het grafische werk kan men kiezen tussen een plotter en een printer. Plotters lenen zich het best voor het afdrukken van vektorgrafieken.
Dotmatrix printers zijn geschikt als het alleen om bitmap grafieken gaat.

Netvoeding

Uitleg is eigenlijk niet nodig want iedereen kent het gebruik van deze connector wel. The Amiga needs some power dus een aansluiting op deze connector ist immer noodzakelijk.

RGB en monochrome aansluiting

Deze connectoren zijn voor de aansluiting van een monitor of televisie. Gebruikt men een RGB-monitor dan komt deze aan de RGB-connector. Bij een monochrome monitor zal de aansluiting plaatsvinden op de monochrome connector. RGB-monitoren zijn verkrijgbaar in digitale-en analoge uitvoering. De meeste analoge monitoren hebben een weergave van 4096 kleuren met een resolutie die 640 x 512 pixels omvat. De digitale monitoren kunnen meestal maar 16 kleuren aan. Een analoge monitor is dus dan de beste keuze. Monochrome monitoren werken maar met één kleur en zijn dus niet echt geschikt voor een gemiddelde AMIGA-gebruiker. De meeste RGB-monitoren beschikken over een knop om over te schakelen van kleur naar monochrome weergave. Dus wie toch op een monochroom beeldscherm wil werken kan dus ook op een RGB monitor uit de voeten. De T.V. is ook aansluitbaar maar dan wel doormiddel van een modulator (A521) die op de RGB-connector wordt aangesloten. Echt handig is deze oplossing niet omdat het beeld niet de perfecte weergave van een RGB-monitor kan evenaren en u meestal niet de enige bent die gebruik maakt van de televisie.
Verder is er nog meer randapparatuur die u op de connectoren kunt aansluiten. Hierop zullen we deze keer niet verder op ingaan.

Tot slot

Reacties, vragen, en ideeën zijn altijd welkom. Dus wilt u dat wij iets gaan behandelen wat in deze beginners rubriek past stuur het ons op onder vermelding van AMIGA STARTERS. De volgende keer zullen we de draad weer verder oppikken met preferences. See you then!

Johan & Johan